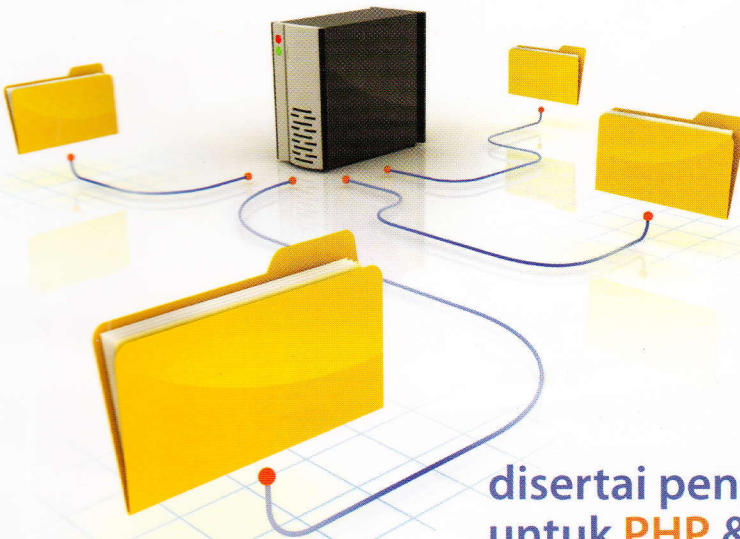




Panduan Belajar

MySQL

Database Server



disertai penggunaannya
untuk **PHP & Visual Basic**

.....

Menginstal MySQL di Windows dan Linux ►

Memadukan MySQL dengan PHP ►

Memadukan MySQL dengan Visual Basic ►

Mengelola User di MySQL ►

Mengoptimalkan MySQL ►

.....

Panduan Belajar

MySQL Database Server

Disertai Penggunaannya untuk PHP dan Visual Basic

Panduan Belajar
MySQL Database Server

Disertai Penggunaannya untuk PHP dan Visual Basic

Wahana Komputer

Panduan Belajar MySQL Database Server

Penulis: **Wahana Komputer**

Penyunting: **Sudarma S.**

Desain Cover: **Budi Setiawan**

Penata Letak: **Imam**

Diterbitkan pertama kali oleh: mediakita

Ilustrasi cover: © **Ahmad Hamoudah**
diperoleh secara legal dari www.istockphoto.com

Redaksi:

Jl. Haji Montong No. 57 Ciganjur Jagakarsa
Jakarta Selatan 12630
Telp. (Hunting): (021) 788 83030; Ext.: 213, 214, 215, 216
Faks. (021) 727 0996
E-mail: redaksi@mediakita.com
Situs web: www.mediakita.com

Pemasaran:

PT TransMedia
Jl. Moh. Kahfi II No.12 A
Cipedak, Jagakarsa, Jakarta Selatan
Telp. (Hunting): (021) 7888 1000
Faks. : (021) 7888 2000
E-mail: pemasaran@transmediapustaka.com

Cetakan pertama, 2010

Hak cipta dilindungi Undang-undang

Katalog Dalam Terbitan (KDT)

Wahana Komputer

Panduan belajar mysql database server / **Wahana Komputer**; penyunting, Sudarma S.;

—cet.1— Jakarta: mediakita, 2010

X +178 hlm.; 15x23 cm

ISBN 979-794-243-0

1. Komputer

II. Sudarma S.

I. Judul

000

Apabila Anda menemukan kesalahan cetak dan atau kekeliruan informasi pada buku ini,
harap menghubungi redaksi mediakita.

Kata Pengantar

MySQL merupakan sebuah software sistem manajemen database yang banyak digunakan dalam praktik, baik di dunia ataupun di Indonesia. Keunggulan MySQL dibandingkan database lain adalah free dan open source.

Adanya software MySQL yang gratisan tentu membuat banyak pihak menyukainya dan mulai menggunakannya. Aplikasi ini selain gratis, tetapi juga andal dan dapat digunakan untuk berbagai platform komputer. Misalnya Linux, Windows, dan MacOS.

Buku *Panduan Belajar MySQL Database Server* ini akan mengajak Anda belajar pemrograman MySQL untuk mengelola database. Anda akan belajar secara cepat, tetapi aplikatif tentang bagaimana menggunakan MySQL.

APA YANG HARUS ANDA KUASAI?

Buku ini akan membahas penggunaan MySQL secara aplikatif. Anda tidak disyaratkan untuk pernah menggunakan aplikasi ini sebelumnya, tetapi paling tidak Anda sudah menguasai penggunaan sistem operasi, seperti Windows atau Linux.

Daftar Isi

Kata Pengantar	v
Daftar Isi	vi

MENGENAL MySQL	1
APA ITU DATABASE	2
DATABASE RELASIONAL	2
TABEL	4
QUERY	4
MENGENAL MYSQL	5
KELEBIHAN DAN KEUNTUNGAN MEMAKAI MYSQL	6
PERBANDINGAN MEMORI DATABASE MYSQL	
DENGAN DATABASE LAIN	7
BAGAIMANA MYSQL BEKERJA?	8
KEGUNAAN MYSQL	9
BAGAIMANA MEMULAI MySQL	11
PERSIAPAN INSTALASI	12
INSTALASI MYSQL DI WINDOWS	13
INSTALASI MYSQL DI LINUX	19
INSTALASI MYSQL DENGAN CARA MANUAL	19
INSTALASI MYSQL DENGAN CARA OTOMATIS	21

MI MULAI MYSQL DI WINDOWS	21
KELILIN MYSQL DI WINDOWS	23
MI MULAI MYSQL DI LINUX	23
TOOL ADMINISTRASI DAN SECURITY	24
USERNAME DAN PASSWORD PADA MYSQL	25
KI WENANGAN SISTEM MYSQL	25
PASSWORD HASHING	26
KI AMANAN KONEKSI MYSQL	28

MI MBANGUN DATABASE	29
MI MBUAT DAN MENGGUNAKAN DATABASE	30
MI MBUAT TABLE	32
MI MASUKKAN DATA	34
MI LIHAT ISI DATA PADA TABEL	36
PRIMARY KEY	38
UNIQUE KEY	38
FOREIGN KEY	40
MI NAMBAH PRIMARY KEY	41
MI NAMBAH UNIQUE KEY	42
MI NAMBAH FOREIGN KEY	43
MI NGUBAH NAMA TABLE	44
MI NAMBAH KOLOM/FIELD	45
MI NGURANGI KOLOM	46
MI NGHAPUS DATABASE	47
MI NGHAPUS TABEL	48
MI NGHAPUS PRIMARY KEY	50

MENGHAPUS UNIQUE KEY	51
INDEX	52
QUERY	53
STATEMENT SELECT	53
PERINTAH REPLACE	60
MEMPERBARUI DATA (UPDATE)	62
JOIN TABEL	64
CROSS-JOIN	65
STRAIGHT JOIN	67
INNER JOIN	68
LEFT [OUTER] JOIN	68
RIGHT [OUTER] JOIN	69
IDENTIFIER.....	70
PENGEMBANGAN MySQL.....	71
INSTALASI PHP DAN APACHE	72
INTERAKSI DATABASE MYSQL DENGAN PHP	75
MENGAKTIFKAN DATABASE.....	77
MENAMPILKAN ISI DARI DATABASE	79
MEMBUAT PROGRAM HAPUS PADA DATABASE	85
PENGUNAAN PHPMyAdmin UNTUK WEB DEVELOPMENT	95
DASBOR PHPMYADMIN	99
MENGENAL TAB PHPMYADMIN	100
MEMBUAT DAN MENGHAPUS DATABASE	107
MEMBUAT TABEL.....	110

MENGGISI DAN MENGGUBAH DATA DI TABEL	112
MENCARI NILAI TERTENTU	114
MENGHAPUS RECORD.....	118
EKSPOR DAN IMPOR DATA	120
OPERASI TABEL	122
TRUNCATE DAN DROP	123
ODBC	125
INSTALASI ODBC MYSQL CONNECTOR 5.1.5.....	128
MEMBUKA ODBC DATA SOURCE ADMINISTRATOR	130
MEMBUAT USER DSN	133
MENYHUBUNGKAN MICROSOFT ACCESS MYSQL.....	135
OPERASI OFFICE.ORG BASE 3 DENGAN MYSQL	139
MENAMBAHKAN DATA KE MYSQL.....	144
MEMBUAT TABEL DI MYSQL BARU DENGAN BASE.....	145
MANAJEMEN USER.....	151
MENGENAL MODUS AUTENTIFIKASI MYSQL.....	152
MANAJEMEN USER	156
PEMBUATAN USER	156
SETTING PASSWORD	157
GRANT PRIVILEGE.....	158
MENCABUT PRIVILEGE	159
MELIHAT PRIVILEGE USER	161
MENGHAPUS USER	161
MANAJEMEN USER DI PHPMyAdmin	162

OPTIMASI MySQL	167
KONFIGURASI HARDWARE	168
OPTIMASI STRUKTUR TABEL.....	170
PERINTAH EXPLAIN SELECT.....	170
MENGUNAKAN FLUSH	172
MENGUNAKAN SHOW	173
 Penutup.....	 176
Tentang Penulis.....	177

MENGENAL MySQL

Bab ini akan membahas:

- Pengenalan Database, konsep, dan atribut.
- Fungsi serta kelebihan MySQL.

MySQL merupakan database server open source yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh para praktisi untuk membangun suatu project. Adanya fasilitas API (Application Programming Interface) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

APA ITU DATABASE?

Database adalah sebuah struktur yang umumnya terbagi dalam 2 hal, yaitu sebuah database flat dan sebuah database relasional. Database relasional lebih mudah dipahami daripada database flat karena database relasional mempunyai bentuk yang sederhana serta mudah dilakukan operasi data. MySQL sendiri adalah sebuah database relasional. Database yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Pada setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefinisikan jenis informasi apa yang harus disimpan.

Mengapa menggunakan database? Itu pertanyaan yang akan keluar dari pikiran Anda pada saat pertama kali ingin mempelajari database. Database akan menjadi sangat berguna saat Anda perlu menyimpan informasi yang dikategorikan secara logis. Contoh, jika Anda ingin menyimpan informasi tentang PT. Wahana Komputer dengan database, Anda bisa mengelompokkan berbagai hal dalam bisnis menjadi beberapa tabel.

DATABASE RELASIONAL

Ketika Anda menggunakan software sistem manajemen database terkomputerisasi, umumnya menggunakan database relasional. Prinsip database relasional adalah informasi dibagi menjadi beberapa data yang terpisah secara logis. Data-data yang terpisah tersebut diletakkan dalam

bentuk tabel. Tabel adalah objek dasar yang merupakan jantung dari database relasional. Tabel adalah dasar penyimpanan informasi dan pengambilannya (retrieval). Ketika informasi sudah tersimpan di tabel-tabel yang terpisah, Anda nantinya dapat melihat (view), mengedit (edit), menambah (add), dan menghapus (delete) informasi dengan berbagai metode. Selain itu, Anda juga bisa mengambil informasi menggunakan query dan menampilkan informasi menggunakan report.

Keunggulan penyimpanan data menggunakan database relasional sangat banyak dibandingkan dengan penyimpanan ke satu tabel ukuran besar dua dimensi (yang disebut file flat) seperti di dokumen Word atau spreadsheet Excel.

Salah satu keunggulan utama database adalah berkurangnya redundansi data. Konsekuensinya tidak hanya ruang penyimpanan harddisk menjadi berkurang, tetapi kecepatan pemrosesan data juga berkurang. Selain itu, ada kelebihan lainnya, yaitu:

- **Fleksibilitas:** jika data berubah, Anda dapat meng-update nilainya hanya di satu tempat. Dengan demikian, semua query, form, dan lainnya akan berubah secara konsekuen dengan nilai yang baru tersebut.
- **Simple:** model penyimpanan tabel yang merupakan dasar dari sistem relasional memang simple, dan merupakan metode penyimpanan data yang tak redundan. Tiap tabel didesain relasional untuk objek tunggal yang mengandung data, yang konsekuen terhadap aspek tertentu dari database, seperti pegawai, produk, atau order.
- **Power:** menyimpan data dalam tabel-tabel yang terpisah lebih memudahkan adanya pengelompokan, searching, dan pengambilan data menggunakan banyak cara yang tak terbatas.
- **Pemudahan manajemen:** dengan tabel yang kecil dan tidak kompleks, informasi jadi mudah dilacak dan diatur.

Misalnya, Anda menggunakan database untuk menyimpan data order dan pelanggan. Jika menggunakan tabel tunggal, ada banyak record yang dibuat untuk tiap order yang dilakukan pelanggan. Dengan demikian, tiap kali ada order, tiap kali pula sebuah informasi dibuat, walaupun pelanggannya sama. Karena itu, jika suatu saat ada informasi

yang berubah (misalnya informasi pelanggan), semua record yang mengandung informasi pelanggan harus diubah. Ini akan sangat merepotkan. Begitu pula informasi tentang pegawai pemroses order juga akan terus berulang.

Akan lebih efisien seandainya ada satu tabel untuk pelanggan dan satu tabel untuk pegawai. Kemudian, ada field identitas pelanggan yang di-tambahkan ke tabel pelanggan dan tabel order sehingga ada hubungan/koneksi antar keduanya. Hubungan ini disebut *relationship*. Begitu pula identitas pegawai bisa dihubungkan dengan tabel order menggunakan *relationship*. Secara lengkap, komponen utama database dijelaskan seperti berikut ini.

TABEL

Tabel adalah inti konsep database. Tujuannya adalah menyimpan informasi. Tabel satu dengan lainnya bisa dihubungkan. Satu database bisa mengandung banyak tabel dalam jumlah tak terbatas. Jumlah record dalam tabel juga umumnya tak dibatasi, batasnya hanya pada kapasitas disk yang digunakan. Tiap tabel memiliki beberapa kolom, kolom menentukan nilai-nilai apa yang bisa disimpan. Misalnya, tabel yang menyimpan informasi produk perlu menyimpan data seperti nama, harga, dan berat produk.

Setiap baris dari tabel bisa mengandung nilai untuk kolom-kolom yang ditentukan di tabel. Selain itu, tiap kolom bisa ditentukan tipe data apa saja yang didukung. Tipe data menentukan jenis data yang bisa disimpan. Tipe data bisa dibatasi seperti jenis numerik atau tanggal, atau lain sebagainya.

QUERY

Anda dapat mengambil informasi tertentu yang disimpan di tabel atau multitabel menggunakan sebuah query. Untuk membuat query, Anda harus menentukan parameter-parameter dari informasi yang ingin dicari.

Misalnya, Anda ingin mengambil data pelanggan yang telah membeli barang tipe X selama 3 bulan yang lalu. Dengan demikian, Anda harus mengisikan variabel nama barang dan periode. Informasi yang ditampilkan juga nantinya bisa diurutkan, di-filter dan diatur cara menampilkannya.

Dengan menjalankan query, Anda bisa menampilkan data berupa nilai-nilai yang sesuai dengan kriteria.

MENGENAL MYSQL

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius, yang dapat Anda hubungi di-emailnya monty@analytikerna.

MySQL database server adalah RDBMS (Relasional Database Management system) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut resource yang besar. MySQL adalah database yang paling populer diantara database-database yang lain.



Gambar 1.1 Logo MySQL.

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware. Penulis sendiri dalam menjelaskan buku ini menggunakan MySQL yang free software karena bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (general public license), yang dapat Anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut:

- MySQL dirilis pertama kali secara internal pada 23 Mei 1995.
- Versi Windows dirilis pada 8 Januari 1998 untuk Windows 95 dan Windows NT.
- Versi 3.23: beta dari Juni 2000, dan dirilis pada January 2001.
- Versi 4.0: beta dari Agustus 2002, dan dirilis pada maret 2003 (unions).
- Versi 4.1: beta dari bulan Juni 2004, dirilis pada bulan oktober. 2004 (R-trees dan B-trees, subqueries, prepared statements).
- Versi 5.0: beta dari bulan Maret 2005, dirilis pada Oktober 2005. (cursor, stored procedure, trigger, views XA transaction).
- Sun Microsystems membeli MySQL AB pada tanggal 26 Februari 2008.
- Versi 5.1: dirilis 27 November 2008 (event scheduler, partitioning, plug-in API, row-based replication, server log table).

KELEBIHAN DAN KEUNTUNGAN MEMAKAI MYSQL

Jika Anda mencari system manajemen database, ada beberapa pilihan, antara lain:

- MySQL
- Oracle
- PostgreSQL
- mSQL
- Microsoft SQL Server

Ketika dibandingkan antara MySQL dengan system manajemen database yang lain, perlu diperhitungkan apa yang paling penting untuk Anda. Apakah performa, support, fitur-fitur SQL, kondisi keamanan dalam license, atau masalah harga. Dengan pertimbangan tersebut, database

MySQL memiliki beberapa kelebihan dan keuntungan dibanding database lain, di antaranya adalah:

- Banyak ahli berpendapat MySQL merupakan server tercepat.
- MySQL merupakan system manajemen database yang OpenSource (kode sumbernya terbuka), yaitu software ini bersifat free atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
- MySQL mempunyai performa yang tinggi tapi simpel.
- Database MySQL mengerti bahasa SQL (Structured Query Language).
- MySQL dapat diakses melalui protocol ODBC (Open Database Connectivity) buatan Microsoft. Ini menyebabkan MySQL dapat diakses oleh banyak software.
- Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses database.
- Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
- MySQL merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran Gigabyte.
- MySQL dapat berjalan di berbagai operating system seperti Linux, Windows, Solaris, dan lain-lain.

PERBANDINGAN MEMORI DATABASE MYSQL DENGAN DATABASE LAIN

Setiap database mempunyai batasan-batasan memori yang berbeda-beda. Berikut ini adalah perbandingan batasan memori MySQL dengan database-database yang lain.

Tabel 1.1 Ukuran DB Max beberapa software RDBMS

Database	Ukuran DB Max
4th Dimension	Tak terbatas
Advantage Database Server	Tak terbatas
Datawasp	Tak terbatas
DB2	512 TB
Firebird	Tak terbatas
Ingress	Tak terbatas
Microsoft Access	2 GB
Microsoft Visual FoxPro	2 GB
Microsoft SQL Server	524.258(32.767 file * 16 TB ukuran file maksimal)
MySQL 5	Tak terbatas
Oracle	Tak terbatas (4 GB * ukuran blok pertablespace)
OpenEdge	sekitar 32 exabytes
Polyhedra DBMS	Dibatasi oleh RAM, address space
PostgreSQL	Tak terbatas
Teradata	Tak terbatas

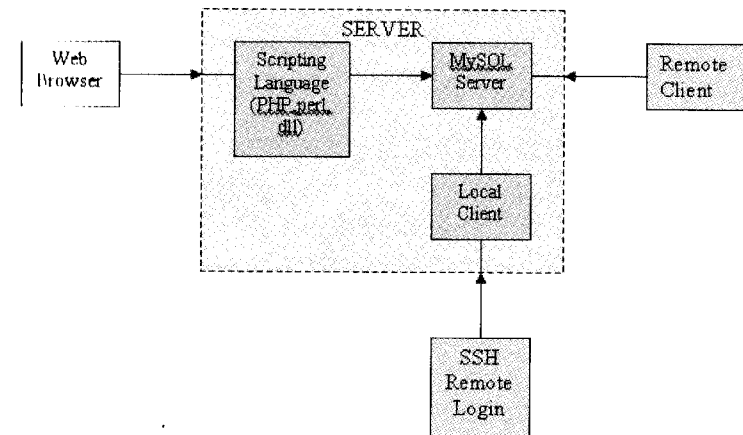
BAGAIMANA MYSQL BEKERJA?

Keunggulan MySQL adalah kemampuannya dalam menyediakan berbagai fasilitas atau fitur-fitur yang dapat digunakan oleh bermacam-macam user. User ini sendiri termasuk administrator database, programmer aplikasi, manager, sampai end user (pemakai akhir).

Pada mulanya MySQL bekerja pada platform unix and linux. Namun, dengan perkembangannya, sekarang banyak bermunculan beberapa distro yang mampu berjalan pada beberapa platform yang bersifat shareware dan corporate.

MySQL adalah sebuah manajemen system database server yang mampu menangani beberapa user, yaitu mampu menangani beberapa instruksi sekaligus dari beberapa user dalam satu waktu. Dan, MySQL merekam semua data user di dalam sistemnya dalam tabel user.

Untuk pengguna yang menjalankan MySQL pada platform Windows



Gambar 1.2 Prototype server MySQL.

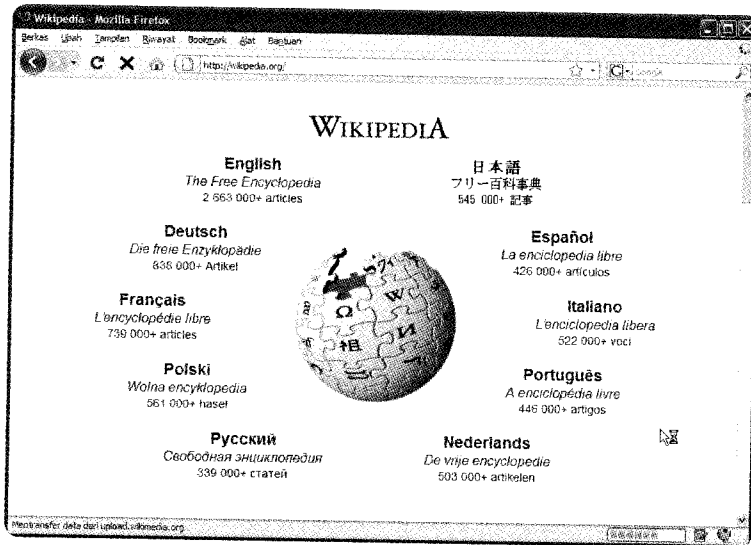
dapat mengakses melalui program client seperti putty, telnet, psftp, dan lain lain. Sedangkan untuk pengguna yang menjalankan MySQL pada platform linux dapat mengakses melalui program client seperti SSH maupun telnet, seperti gambar 1.2 di atas yang menggambarkan server MySQL dengan aplikasi server side dan SSH.

KEGUNAAN MYSQL

MySQL sangat populer utamanya untuk aplikasi web. MySQL merupakan komponen database "M" yang ada di LAMP, BAMP, MAMP, dan WAMP (Linux/BSD/Mac/Windows-Apache-MySQL-PHP/Perl/Python). Selain itu, digunakan di tool bugtracking seperti Bugzilla.

Kenapa MySQL populer? Beberapa penyebabnya antara lain adalah MySQL sangat disukai oleh pengembang bahasa pemrograman untuk web, seperti PHP dan RoR (ruby on rails).

MySQL terbukti kestabilannya karena digunakan oleh banyak proyek besar, seperti MediaWiki, Flickr, facebook, Google, Nokia dan Youtube. Selain itu, proyek CMS free seperti WordPress, Drupal, Joomla semuanya menggunakan MySQL sebagai database server-nya



Gambar 1.3 Wikipedia, salah satu proyek open source yang menggunakan MySQL sebagai database.

BAGAIMANA MEMULAI MySQL

Bab ini akan membahas:

- Persiapan instalasi MySQL.
- Instalasi MySQL pada platform Windows.
- Instalasi MySQL pada platform Linux.
- Cara memulai MySQL.
- Administratif tool dan security.

Pada bab ini akan dibahas tentang persiapan instalasi MySQL baik pada platform Windows maupun pada platform Linux. Untuk instalasi baik pada platform Windows maupun platform Linux dilakukan secara default. Anda bisa mengganti susunannya tanpa harus sama persis atau mengubah susunannya sesuai kebutuhan database yang Anda perlukan.

Anda dapat menginstal dengan cara manual ataupun dengan otomatis menggunakan distribusi RPM. Pada Windows, Anda dapat menjalankannya dengan beberapa cara. Anda dapat menjalankan MySQL server melalui command prompt maupun dari command line client. Di sesi ini juga dibahas tentang administrative tool dan security, yang meliputi username dan password yang secara default username-nya adalah root dan password-nya adalah password pada saat penginstalan. Anda juga dapat membuat username dan password yang baru. Username dan password yang baru akan disimpan pada tabel user pada database MySQL atau disebut password hashing yang akan dijelaskan.

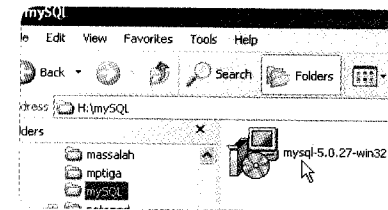
PERSIAPAN INSTALASI

Tentu yang pertama Anda persiapkan adalah sebuah komputer yang memakai platform Windows minimal Windows 95. MySQL sendiri terdapat 2 format distribusi yaitu format binary yang berisi program setup dan format source code yang memakai VC++ 6.0 compiler. Begitu juga yang berjalan pada platform linux memiliki distribusi yang berbentuk binary, RPM, dan source format. Binary dan RPM lebih mudah untuk diinstal, sedangkan yang berbentuk source format lebih sulit untuk diinstal karena Anda harus mengompilasi software tersebut. Anda bisa mendownload MySQL sesuai sistem operasi yang Anda gunakan di sini: <http://dev.mysql.com/downloads/mysql/>.

INSTALASI MYSQL DI WINDOWS

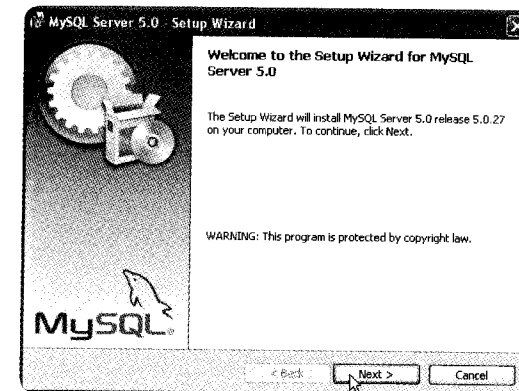
berikut ini adalah langkah-langkah menginstal MySQL pada Windows.

1. Buka folder yang berisi instalasi MySQL (misal, **H:\MySQL**) lalu klik ganda **mysql-5.0.27-win32.exe**.



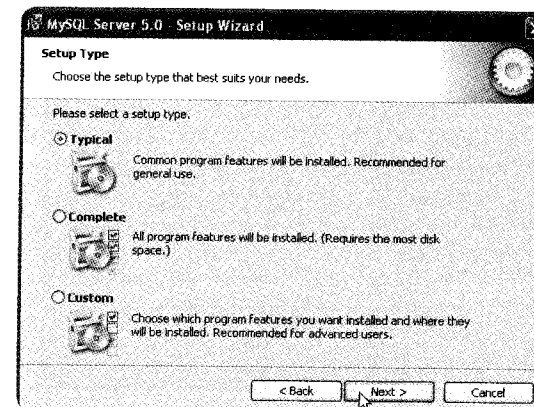
Gambar 2.1 Folder berisi installer MySQL.

2. Klik **Next**.



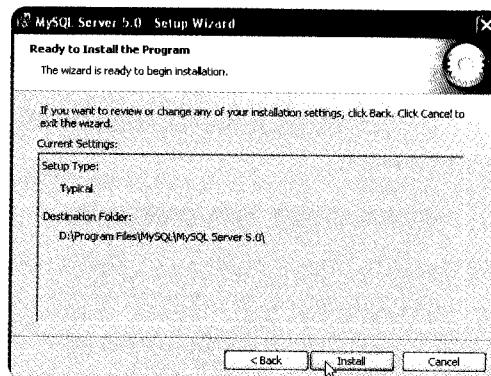
Gambar 2.2 Tampilan pertama setup wizard.

3. Pilih **Typical**, lalu klik **Next**.



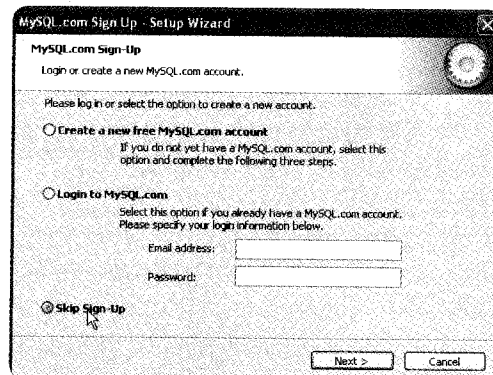
Gambar 2.3 Kotak dialog Setup Type.

4. Klik **Install**.



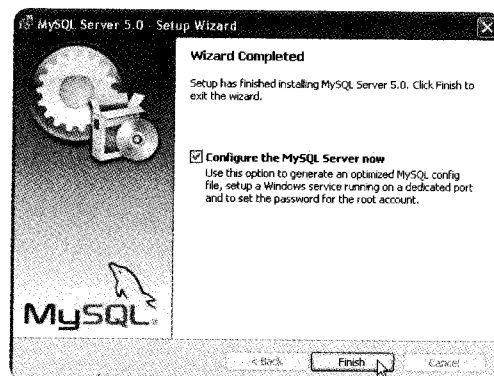
Gambar 2.4 Memulai instalasi.

5. Pilih **Skip Sign-Up**, lalu klik **Next** untuk melewati kotak dialog MySQL.com sign-up.



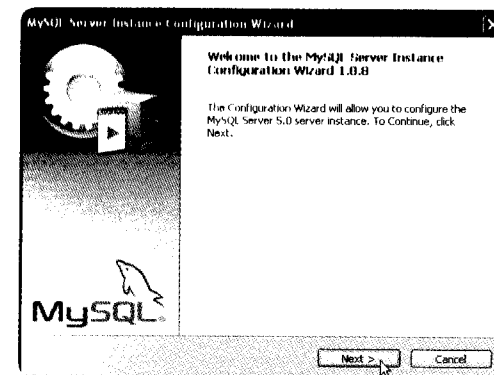
Gambar 2.5 Kotak dialog MySQL.com Sign Up.

6. Ceklis **Configure the MySQL Server now** lalu klik **Finish**.



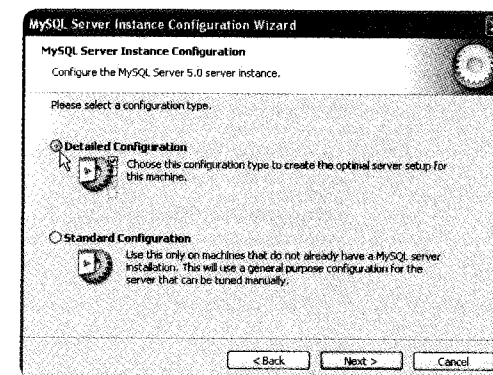
Gambar 2.6 Instalasi selesai.

7. Klik **Next**.



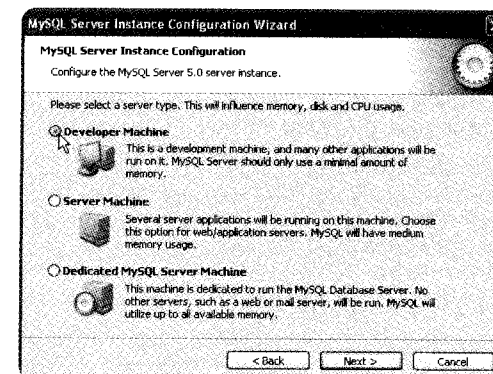
Gambar 2.7 Wizard konfigurasi.

8. Pilih **Detailed Configuration**, lalu klik **Next**.



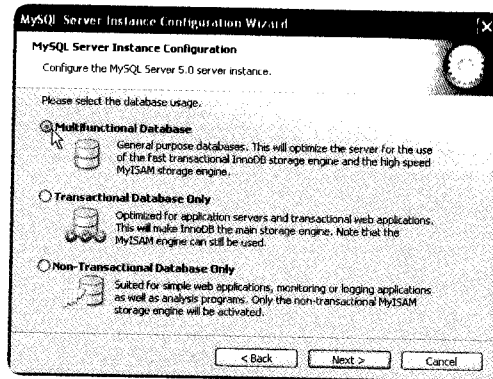
Gambar 2.8 Tipe konfigurasi.

9. Pilih **Developer Machine**, lalu klik **Next**.



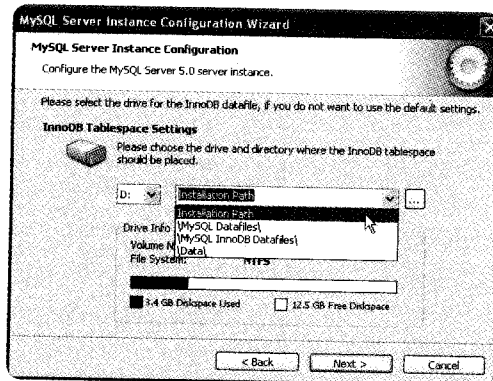
Gambar 2.9 Tipe server.

10. Pilih **Multifunctional Database**, lalu klik **Next**.



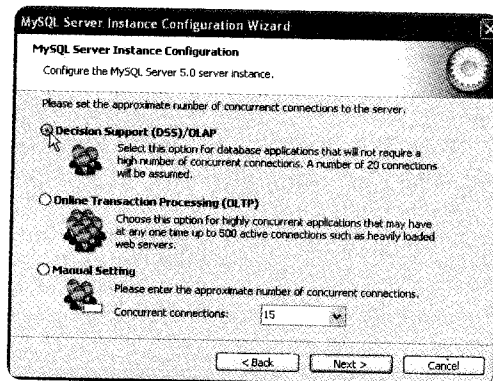
Gambar 2.10 Penggunaan MySQL.

11. Pilih **Installation Path**, lalu klik **Next**.



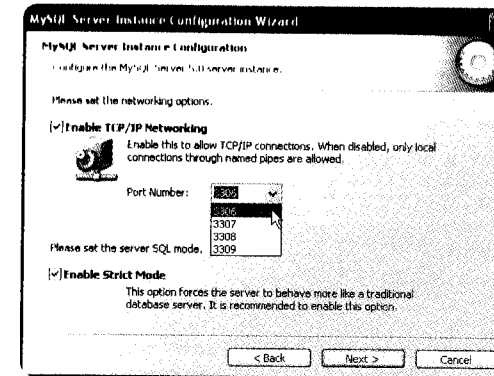
Gambar 2.11 Drive untuk lokasi file data InnoDB.

12. Pilih **Decision Support (DSS)/OLAP**, lalu klik **Next**.



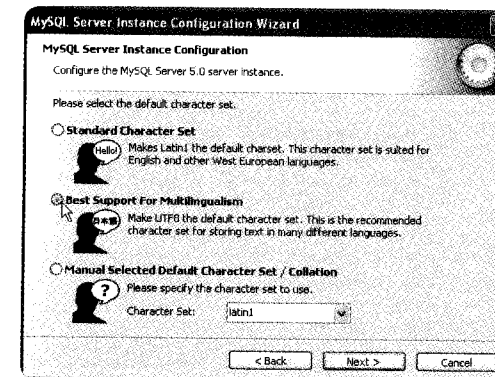
Gambar 2.12 Jumlah maksimal koneksi yang bersamaan ke server.

13. Pilih **Enable TCP/IP Networking** dan **Enable Strict Mode**, pilih Port Number **3306**. Lalu, klik **Next**.



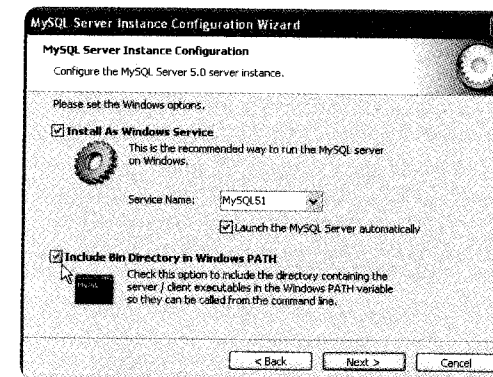
Gambar 2.13 Opsi jaringan.

14. Pilih **Best Support For Multilingualism**, lalu klik **Next**.



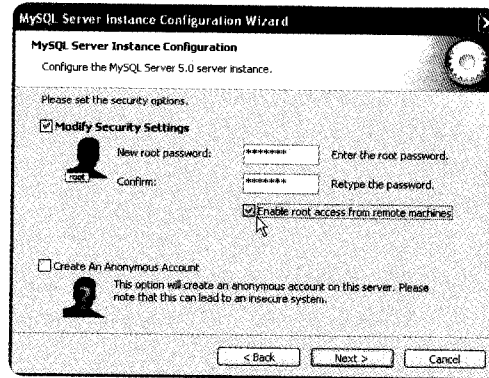
Gambar 2.14 Karakter yang didukung.

15. Pilih **Service Name MySQL51**, lalu klik **Next**.



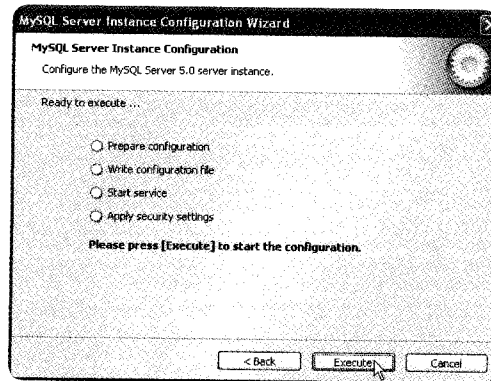
Gambar 2.15 MySQL sebagai service Windows.

16. Isi password Anda, lalu klik **Next**.



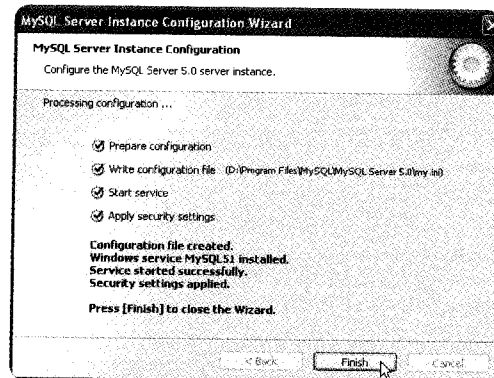
Gambar 2.16 Password untuk keamanan.

17. Klik **Execute**.



Gambar 2.17 Menjalankan perubahan konfigurasi.

18. Klik **Finish**.



Gambar 2.18 Konfigurasi MySQL telah selesai.

INSTALASI MYSQL DI LINUX

Dalam Linux, MySQL dapat Anda install dalam dua cara, yaitu manual maupun otomatis dengan menggunakan RPM.

INSTALASI MYSQL DENGAN CARA MANUAL

Distribusi dalam bentuk source format biasanya bernama `mysql-versi.tar.gz`. Versi adalah nomor versi MySQL. Berikut ini adalah langkah-langkah instalasi MySQL pada Linux:

1. Ekstrak source MySQL.

```
# tar xvfz mysql-versi.tar.gz
```

2. Pindah ke direktori MySQL.

```
# cd mysql-versi
```

3. Lakukan konfigurasi sistem sebelum proses kompilasi dilakukan.

```
# ./configure
```

Jika Anda ingin menggunakan option untuk configure, Anda dapat melihat daftarinya dengan perintah:

```
# ./configure --help
```

4. Beberapa option yang biasa digunakan:

`--without server`

5. Konfigurasi menjadikan komputer Anda sebagai klien dari server yang sudah ada pada komputer lain.

`--prefix=nama_path`

6. Biasanya terletak di direktori `/usr/local/mysql`. Data direktori, klien, server, klien library, dan file header diinstal pada direktori `var`, `bin`, `libexec`, `lib` dan `include` di bawah direktori ini. Jika Anda ingin mengganti lokasinya, menggunakan option sebagai berikut:

`--localstatedir=nama_path`

Default-nya terletak di direktori `/usr/local/mysql/var`. Dengan option ini, Anda dapat mengganti letaknya.

`--with-low-memory`

File source `sql/sql_yacc.cc` membutuhkan memori yang besar dalam proses kompilasi sehingga terkadang akan menyebabkan proses error karena sistem kehabisan virtual memori. Dengan option ini memori akan diperkecil.

Compile

```
# make && make install
```

Setelah proses compile selesai, seluruh file binary, dokumentasi dan segala macamnya mengenai MySQL telah diletakkan pada direktori default (`/usr/local/mysql`).

Untuk memulai MySQL, lakukan instalasi database (grant table) default.

```
# cd /usr/local/mysql/bin
```

```
# ./mysql_install_db
```

Lalu jalankan MySQL.

```
# ./safe_mysqld &
```

Lalu cek pada proses komputer, apakah MySQL sudah berjalan.

```
# ps ax|grep sql
```

Akan tampak hasil sebagai berikut:

```
557 ? S      0:00 sh/usr/bin/safe_mysqld
--user=mysql --pid-file=/var/
```

```
587 ? SN    0:00 /usr/sbin/mysqld --basedir=/
--datadir=/var/lib/mysql
```

INSTALASI MYSQL DENGAN CARA OTOMATIS

Instalasi dengan format RPM ada beberapa macam, antara lain:

`MySQL-versi-platform.rpm` untuk server MySQL.

`MySQL-client-versi-platform.rpm` untuk klien MySQL.

`MySQL-devel-versi-platform.rpm` software pendukung untuk mengembangkan klien MySQL berisi klien library dan file header.

`MySQL-bench-versi-platform.rpm` untuk benchmark dan test MySQL.

`MySQL-versi.src.rpm` untuk semuanya yaitu source lengkap untuk server, klien, benchmark, dan test.

Anda tidak perlu menginstal semua source RPM yang ada, cukup yang Anda butuhkan saja. Berikut adalah beberapa pilihan yang bila Anda ingin menginstal source RPM, antara lain:

Bila Anda berencana menginstal server sekaligus klien, Anda gunakan `mysql`

```
# rpm -ivh MySQL-versi-platform.rpm MySQL-
client-versi-platform.rpm
```

Bila Anda ingin menginstal MySQL dari source lengkapnya, Anda gunakan script:

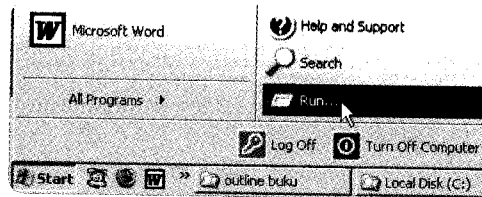
```
#rpm --recompile MySQL-versi.src.rpm
```

MEMULAI MYSQL DI WINDOWS

Untuk memulai MySQL pada Windows ada dua cara, yaitu menggunakan command prompt dan menjalankan klien MySQL.

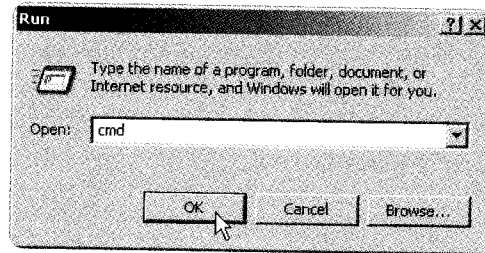
Untuk menjalankan dengan command prompt :

1. Klik **Start**, pilih **Run**.



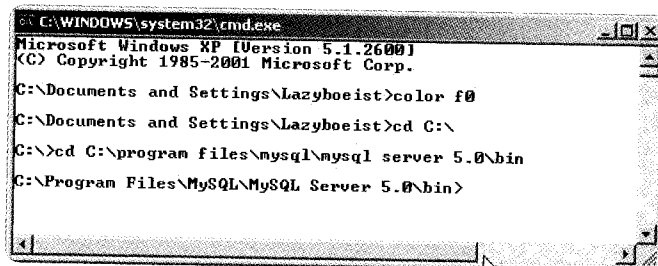
Gambar 2.19 Menjalankan Run.

2. Ketik **cmd** lalu klik **OK**.



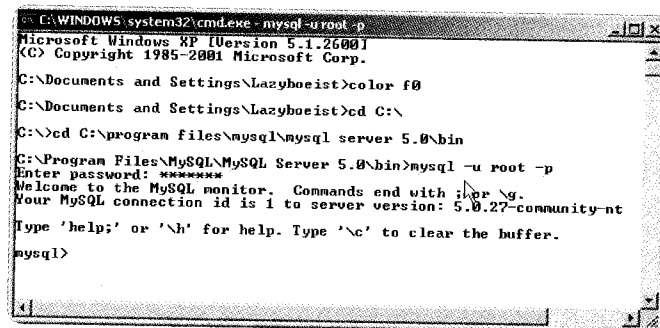
Gambar 2.20 Kotak Run untuk masuk ke Command Prompt.

3. Ketik **cd C:\program files\mysql\mysql server 5.0\bin**.



Gambar 2.21 Direktori file mysql.exe berada.

4. Ketik **mysql -u root -p** lalu tekan **Enter**, masukkan password pada waktu penginstalan MySQL pertama kali.

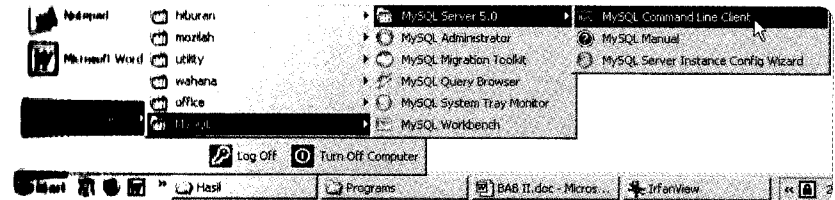


Gambar 2.22 Masuk ke server MySQL

KLIEN MYSQL DI WINDOWS

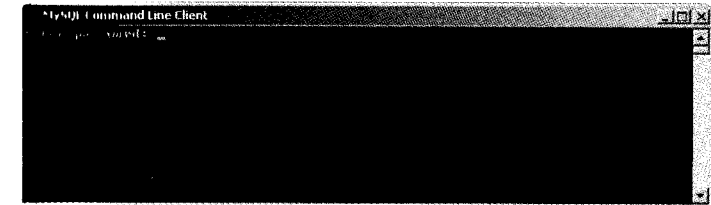
Untuk menjalankan klien MySQL adalah sebagai berikut:

1. Klik **Start > All Programs > MySQL > MySQL Server 5.0 > MySQL Command Line Client**.



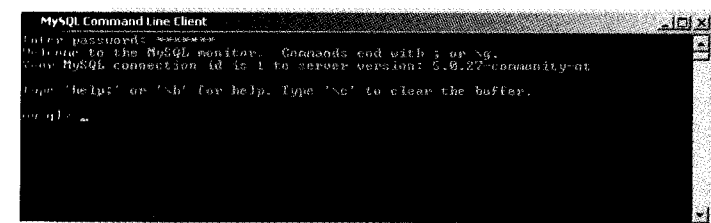
Gambar 2.23 Menu MySQL Command Line Client pada Menu Start.

2. Isikan password yang Anda isikan pada waktu proses instalasi berlangsung.



Gambar 2.24 Permintaan Password.

3. Setelah password diisi dengan benar, akan tampil seperti gambar berikut.



Gambar 2.25 Tampilan awal MySQL Server.

MEMULAI MYSQL DI LINUX

Untuk menjalankan server secara manual (dengan user=mysql), dapat dilakukan dengan perintah:

Untuk menjalankan server secara manual (dengan user=mysql), dapat dilakukan dengan perintah:

```
$ /usr/bin/mysqladmin -u root -h localhost status
```

Untuk melakukan verifikasi apakah server berjalan dengan baik, dapat dilakukan dengan perintah:

```
$ /usr/bin/mysqladmin -u root version
```

Jika server berjalan, akan tampil:

```
mysqladmin vers 8.14 distrib 4.1.8, for redhat-  
linux-gnu on i386
```

```
copyright (C) 2000 MySQL AB & MySQL finland AB  
& TCX Datakonsult AB this software comes with  
ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to modify and redistribute  
it under the GPL license
```

```
server version      4.1.8-standard
```

```
protocol version    10
```

```
connection          localhost via UNIX socket
```

```
UNIX socket         /var/lib/mysql/mysql.sock
```

```
Uptime:             3 min 14 sec
```

```
Threads: 1 Questions: 2 Slow Queries: 0 Opens:  
6 Flush tables: 1
```

```
Open table : 0 Queries per second avg: 0.002
```

TOOL ADMINISTRASI DAN SECURITY

Tool Administratif dan keamanan mencakup username dan password, bagaimana privilege sistem bekerja, password hashing, dan bagaimana koneksi yang aman.

USERNAME DAN PASSWORD PADA MYSQL

Account pada MySQL didasarkan pada username and host dari client. Account juga memiliki password untuk dapat mengakses database secara full. Berikut ini merupakan ketentuan username dan password pada MySQL, yaitu:

- Username yang digunakan pada MySQL dipakai untuk otentifikasi, tetapi tidak berhubungan dengan username yang digunakan pada Windows atau unix. Pada linux, secara default client dari MySQL menggunakan username pada linux, tetapi hal ini bisa diubah dengan mudah.
- Username pada MySQL bisa menggunakan hingga 16 karakter. Batasan ini merupakan hard-coded di dalam MySQL server dan MySQL client sehingga tidak bisa diubah oleh siapapun.
- Password dari MySQL sama sekali tidak berhubungan dengan password dari OS.
- Enkripsi pada password MySQL menggunakan sebuah algoritma tertentu dan berbeda dengan yang digunakan oleh linux, Enkripsi pada password MySQL diimplementasikan dengan fungsi PASSWORD() SQL. Sementara pada linux enkripsi Password diimplementasikan dengan fungsi ENCRYPT() SQL. Pada MySQL versi 4.1 atau lebih tinggi, proteksi yang digunakan lebih aman daripada yang digunakan sebelumnya.

KEWENANGAN SISTEM MYSQL

MySQL mempunyai prosedur keamanan non-standar, salah satunya adalah privilege system (pengaturan hak-hak pengguna). Fungsi utama dari sistem ini adalah melakukan pengecekan user yang mengakses dari sebuah host dan memberikan sebuah atau beberapa hak seperti SELECT, INSERT, UPDATE, DELETE.

Privilege system memastikan pengguna hanya bisa melakukan operasi tertentu. Ketika seorang user mengakses server MySQL, sistem akan mencari identitasnya dengan melihat host dan username yang dipakai olehnya. Setelah itu, sistem akan menentukan apa saja yang bisa dia lakukan (privilege).

MySQL melihat kedua hal tersebut karena mungkin saja ada user bernama Adit yang mengakses dari office.adcom.com tapi ada user lain yang juga bernama Adit yang mengakses dari home.adcom.com. MySQL meng-handle masalah ini dengan memisahkan user yang berbeda host tapi memiliki username yang sama. Kedua user tersebut dapat memiliki privilege yang berbeda.

MySQL access control melakukan dua tahap ketika sebuah client program yang mengakses server, yaitu:

1. Server melihat apakah client tersebut memiliki hak untuk mengakses server tersebut.
2. Server melihat apa saja yang bisa dilakukan untuk client tersebut untuk memastikan client tidak melakukan illegal operation.

PASSWORD HASHING

User account pada MySQL disimpan pada tabel user pada database mysql. Masing-masing account menunjuk pada satu password pada kolom password. Data password yang disimpan tidak berupa plaintext tapi berupa hash value yang didapat dari fungsi PASSWORD().

MySQL menggunakan password dalam dua fase pada komunikasi client/server:

- Ketika sebuah client mengakses server, ada langkah awalan di mana password yang diberikan oleh client memiliki hash value yang sama dengan hash value yang berada di dalam tabel user.
- Ketika client terhubung dengan server, dia bisa (jika mempunyai privilege tersebut) mengubah password untuk account yang ada di tabel user dengan menggunakan fungsi PASSWORD() untuk

mengenerate password hash atau menggunakan GRANT atau SET PASSWORD statements.

Penggunaan password hashing telah diupdate pada MySQL 4.1 untuk memperbaiki keamanan dan mengurangi kemungkinan password diambil oleh user yang tidak berhak. Namun, penggunaan cara ini hanya bisa dipahami oleh MySQL 4.1 ke atas sehingga bisa menyebabkan masalah kompatibilitas. MySQL versi 4.1 ke atas bisa mengakses MySQL versi sebelumnya karena MySQL bersifat backward-compatible, yaitu kompatibel ke belakang. Namun, dari MySQL versi sebelumnya akan menemui kesulitan dalam menghubungi MySQL 4.1 ke atas. Contohnya, sebuah client dengan MySQL 3.23 yang mencoba mengakses server dengan MySQL 5.1 mungkin akan gagal dengan pesan kesalahan sebagai berikut:

```
shell> mysql -h localhost -u root
```

```
Client does not support authentication protocol  
requested
```

```
by server; consider upgrading MySQL client
```

Berikut ini akan membahas mengenai perbedaan cara password hashing pada MySQL versi awal dan versi terbaru.

Sebelum MySQL 4.1, fungsi hashing pada password akan menghasilkan nilai sepanjang 16 bytes.

```
+-----+  
| PASSWORD('mypass') |  
+-----+  
  
| 6f8c114b58f2ce9e |  
+-----+
```

Sementara pada MySQL 4.1, fungsi hashing telah diubah sehingga akan menghasilkan nilai sepanjang 41 bytes.

```
+-----+  
| PASSWORD('mypass') |
```

```
| *6C8989366EAF75BB670AD8FA7A7E01176A95CEFA |
```

Sebuah hasil dari fungsi hash yang baru akan menghasilkan nilai yang berawalan '*'. Password yang lebih panjang tentu saja memiliki properti kriptografi yang lebih baik sehingga lebih aman dibandingkan dengan metode hashing yang sebelumnya diterapkan.

KEAMANAN KONEKSI MYSQL

MySQL mendukung penggunaan koneksi enkripsi yang aman antara server dan client MySQL menggunakan Secure Sockets Layer (SSL) protocol. Konfigurasi standar dari MySQL menerapkan kecepatan yang paling tinggi sehingga koneksi enkripsi tidak digunakan secara default. Menggunakan koneksi enkripsi tentu menggunakan porsi CPU yang besar yang membutuhkan beberapa pekerjaan tambahan sehingga memperlambat kerja MySQL. Untuk aplikasi yang memang membutuhkan pengamanan tingkat tinggi, dibutuhkan komputasi ekstra.

MySQL mengizinkan penggunaan enkripsi secara personal. Koneksi biasa atau koneksi enkripsi bisa dipilih tergantung dari kebutuhan aplikasi secara individu.

SSL adalah sebuah protokol yang menggunakan algoritma enkripsi untuk memastikan data yang ditransfer melalui public network aman. SSL ini memiliki mekanisme untuk mendeteksi perubahan, kehilangan, atau pengiriman ulang data. SSL juga memiliki algoritma yang bisa melakukan verifikasi identitas.

MEMBANGUN DATABASE

Bab ini akan membahas:

- Cara membangun database.
- Cara membuat tabel.
- Cara melihat isi data.
- Editing struktur data.

Pada bab ini akan dibahas mengenai bagaimana membangun database dengan MySQL. Dimulai dengan pembuatan database, pembuatan tabel, hingga menentukan tipe data apa yang harus dipakai pada kolom-kolom di tabel tersebut sehingga Anda dapat memasukkan data dengan tepat. Anda juga akan diberikan contoh bagaimana membuat database.

MEMBUAT DAN MENGGUNAKAN DATABASE

Sebelum Anda membuat tabel, Anda harus terlebih dahulu membuat database. Dalam perintah SQL tidak mengenal huruf besar maupun kecil, jadi Anda dapat mengetikkan dengan huruf besar maupun kecil. Namun, di sini Anda menggunakan huruf besar agar Anda lebih mudah membedakan perintah SQL dengan objek-objek lainnya.

```
CREATE { DATABASE | SCHEMA } [IF NOT EXISTS]
db_name

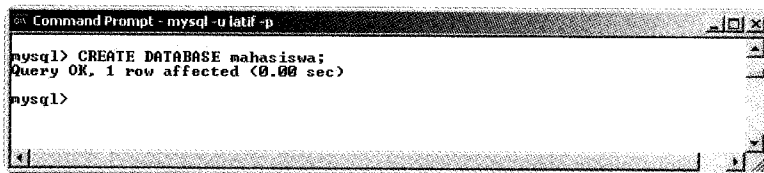
[create_specification [,create_
specification]....]

Create_specification:

[DEFAULT] CHARACTER SET charset_name

| [DEFAULT] COLLATE collation_name
```

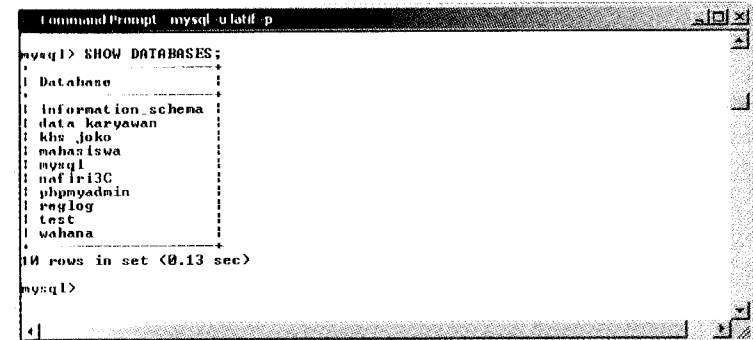
Hal yang perlu Anda perhatikan dalam penamaan database adalah nama database-nya tidak boleh mengandung kata-kata yang menjadi kosakata SQL seperti select. Di samping membuat Anda bingung juga akan membuat rancu. Lihat gambar berikut.



Gambar 3.1 Membuat database baru.

Untuk melihat apakah database sudah benar-benar ada, gunakan perintah berikut:

```
SHOW {DATABASE | SCHEMA} [LIKE 'pattern']
```



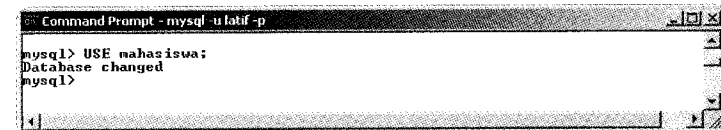
Gambar 3.2 Menampilkan semua database.

Terlihat di atas bahwa database mahasiswa telah berhasil dibuat. Setelah Anda membuat database, jika Anda ingin 'mahasiswa' tersebut menjadi database yang akan Anda akses saat ini, Anda dapat mengakses database tersebut dengan perintah:

```
mysql>USE db_name;
```

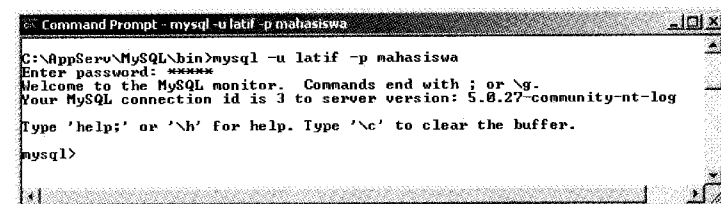
Atau

```
mysql>\u db_name
```



Gambar 3.3 Mengaktifkan database mahasiswa.

Cara di atas digunakan setelah Anda login ke user account. Selain cara tersebut, Anda juga dapat mengakses database tersebut melalui command line.



Gambar 3.3 Mengaktifkan database mahasiswa.

MEMBUAT TABLE

Setelah membuat dan mengaktifkan database, Anda dapat membuat tabel. Sintaks penulisannya sebagai berikut:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table_
name

(create_definition,.....)

[table_option.....]
```

Atau

```
CREATE TABLE nama_table(

kolom1 tipe_data opsi_kolom,

kolom2 tipe_data opsi_kolom,

.....,

kolomN tipe_data opsi_kolom,

PRIMARY KEY (nama_kolom),

INDEX (nama_kolom),

) type=type_tabel;
```

Dalam pemakaian pernyataan CREATE TABLE, Anda akan menemukan 2 opsi yang biasanya menyertai deklarasi definisi kolom, yaitu NULL/NOT NULL dan DEFAULT.

NULL/NOT NULL mengindikasikan apakah suatu kolom dapat menerima nilai null (nilai tiada). Jika dinyatakan null, berarti kolom tersebut bisa dikosongkan. Sebaliknya, bila diberi atribut not null, kolom tersebut tidak boleh kosong. Sedangkan DEFAULT menyatakan nilai baku yang akan dipakai jika dalam suatu proses pemasukan nilai ke kolom tersebut tidak diberi nilai.

Contoh pembuatan tabel mahasiswa dengan kriteria sebagai berikut:

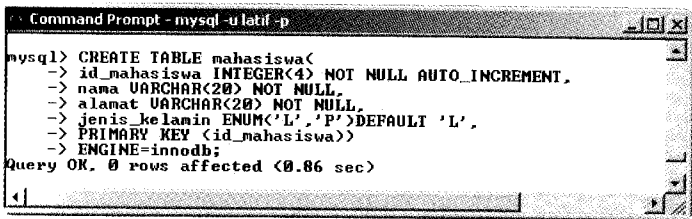
Kolom id_mahasiswa tipe data INTEGER dengan panjang 4 karakter NOT NULL PRIMARY KEY AUTO INCREMENT

Kolom nama tipe data VARCHAR dengan panjang 20 karakter NOT NULL

Kolom alamat VARCHAR dengan panjang 20 karakter NOT NULL

Kolom jenis_kelamin type data ENUM P/L DEFAULT 'L'

TYPE ENGINE innodb

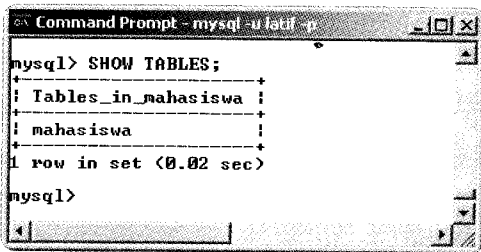


Gambar 3.5 Pembuatan table mahasiswa.

Pada kolom id_mahasiswa NOT NULL ini berarti kolom tersebut tidak boleh kosong, AUTO_INCREMENT ini berarti record nya akan berurutan, PRIMARY KEY ini berarti kunci utama tabel mahasiswa adalah id_mahasiswa. Kolom jenis_kelamin ENUM ini berarti kolom ini hanya bisa diisi data dengan L/P dan apabila Anda tidak mengisinya, secara otomatis akan diisi L(laki-laki).

Setelah Anda membuat tabel, Anda dapat melihat output dari tabel tersebut.

```
SHOW [FULL] TABLES [FROM db_name] [LIKE 'pattern']
```

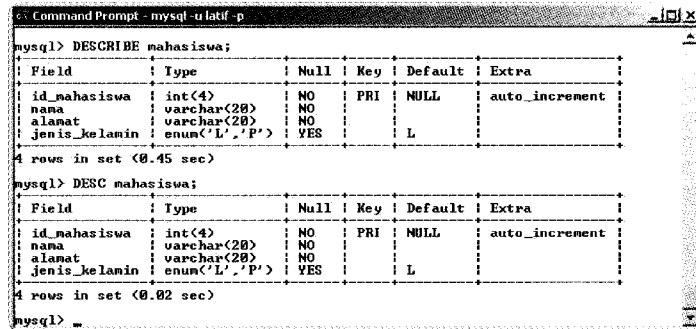


Gambar 3.6 Menampilkan tabel.

Untuk memeriksa tabel yang telah Anda buat sesuai dengan spesifikasi yang telah ditetapkan, gunakan perintah DESCRIBE atau DESC. Sebagai contoh, apabila Anda lupa nama kolom atau tipe kolom dari tabel yang

Anda buat.

```
{DESCRIBE | DESC} table_name [col_name | wild]
```



Gambar 3.7 Spesifikasi kolom.

MEMASUKKAN DATA

Untuk memasukkan data dalam kolom-kolom, Anda dapat menggunakan SQL query INSERT. Dengan perintah ini, Anda dapat memasukkan data yang sesuai dengan spesifikasi dalam kolom-kolom pada tabel.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY]
[IGNORE]
```

```
[INTO] table_name [(colom_name,...)]
```

```
VALUES ({expr | DEFAULT},...), (...),...
```

```
[ON DUPLICATE KEY UPDATE col_name=expr,...]
```

atau

```
INSERT INTO nm_tabel (nama_kol1,nama_kol2,...
nama_koln)
```

```
VALUES
```

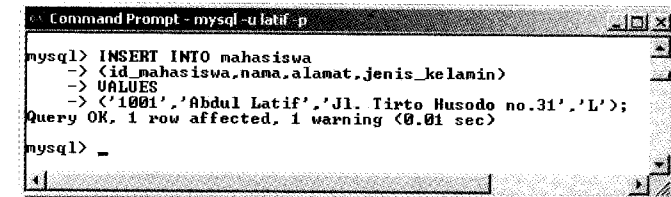
```
('data_kol1','data_kol2',... 'data_koln');
```

Untuk lebih jelasnya, Anda akan memasukkan data ke dalam database mahasiswa pada kolom mahasiswa. Dengan spesifikasi tabel seperti pada gambar.

Field	Type	Null	Key	Default	Extra
id_mahasiswa	int(4)	NO	PRI	NULL	auto_increment
nama	varchar(20)	NO			
alamat	varchar(20)	NO			
jenis_kelamin	enum('L','P')	YES		L	

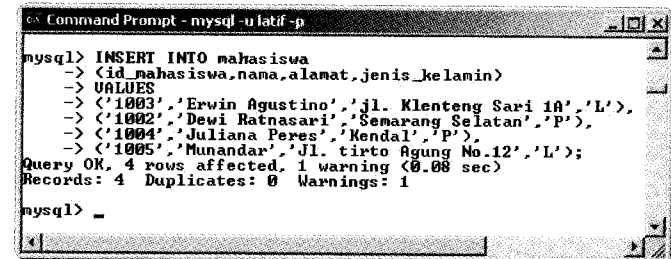
Gambar 3.8 Spesifikasi tabel mahasiswa.

Sebelum Anda memasukkan data ke dalam tabel mahasiswa, perlu Anda perhatikan tipe data yang digunakan dalam setiap kolom.



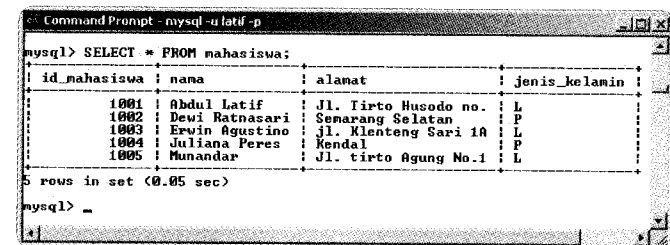
Gambar 3.9 Memasukkan data satu per satu.

Di samping Anda memasukkan data satu per satu, Anda juga dapat memasukkan data secara bersamaan:



Gambar 3.10 Memasukkan data secara bersamaan.

Kita juga dapat melihat data-data yang telah Anda masukkan ke dalam kolom mahasiswa.



Gambar 3.11 Melihat data yang telah dimasukkan.

MELIHAT ISI DATA PADA TABEL

Untuk melihat data yang telah dimasukkan, Anda dapat menggunakan Query SELECT. Dengan menggunakan perintah ini, Anda dapat melihat data dari satu kolom atau beberapa, satu tabel, bahkan beberapa tabel.

SELECT

[ALL | DISTINCT | DISTINCTROW]

[HIGH_PRIORITY]

[STRAIGHT_JOIN]

[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]

[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]

select_expr, ...

[FROM table_references

[WHERE where_condition]

[GROUP BY {col_name | expr | position}

[ASC | DESC], ... [WITH ROLLUP]

[HAVING where_condition]

[ORDER BY {col_name | expr | position}

[ASC | DESC], ...]

[LIMIT {[offset,] row_count | row_count
OFFSET offset}]

[PROCEDURE procedure_name(argument_list)]

[INTO OUTFILE 'file_name' export_options

| INTO DUMPFILE 'file_name'

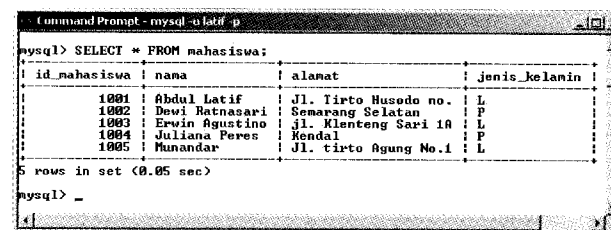
[INTO OUTFILE 'file_name' |, 'file_name']

[FOR UPDATE | LOCK IN SHARE MODE]]

Atau

SELECT FROM table_name;

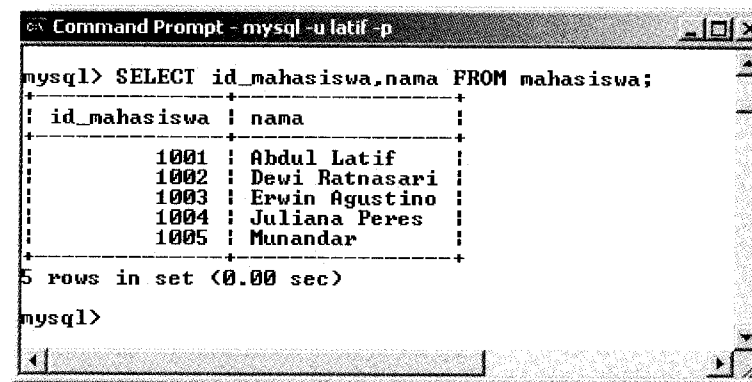
Apabila Anda tidak menginginkan melihat seluruh tabel, hanya be-



Gambar 3.12 Melihat data pada tabel secara keseluruhan.

berapa kolom saja yang Anda butuhkan, Anda pun dapat melakukannya dengan perintah select.

Pada gambar di atas, hanya ditampilkan data pada id_mahasiswa dan



Gambar 3.13 Data pada kolom id_mahasiswa dan nama.

nama saja, tidak keseluruhan dari tabel mahasiswa.

PRIMARY KEY

Primary key merupakan constraint yang diterapkan ke sebuah tabel atau lebih kolom sehingga mempunyai nilai unik dalam mengidentifikasi baris-baris pada suatu kolom. Kolom-kolom tersebut disebut kolom-kolom primary key. Kolom primary key tidak boleh berisi nilai NULL karena digunakan secara unik untuk mengidentifikasi baris-baris pada suatu tabel. Primary key berguna untuk memastikan integritas seluruh data. Untuk lebih jelasnya, lihat contoh berikut.

```
Command Prompt - mysql -u latif -p
mysql> CREATE TABLE karyawan(
-> id INTEGER(4),
-> nama VARCHAR(20),
-> jabatan VARCHAR(20),
-> PRIMARY KEY(id));
Query OK, 0 rows affected (1.27 sec)
```

Gambar 3.14 Penggunaan primary key.

Pada baris paling bawah terlihat bahwa id digunakan sebagai PRIMARY KEY karena kolom id dianggap sebagai kolom yang unik. Hasilnya seperti di bawah ini.

```
Command Prompt - mysql -u latif -p
mysql> desc karyawan;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(4)        | NO   | PRI |          |       |
| nama  | varchar(20)   | YES  |     |          |       |
| jabatan | varchar(20)   | YES  |     |          |       |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)

mysql>
```

Gambar 3.15 Spesifikasi kolom-kolom pada tabel karyawan.

Terlihat di atas pada baris id, kolom KEY terdapat tanda PRI, ini berarti bahwa kolom id digunakan sebagai PRIMARY KEY.

UNIQUE KEY

Unique key digunakan untuk memberikan keunikan pada kolom yang bukan merupakan kolom primary key. Kolom primary key secara otomatis memberi constraint yang unik. Constraint dari unique key hampir

sama dengan primary key. Perbedaannya hanya pada kolom unique key dapat diisi NULL, sedangkan primary key tidak dapat diisi NULL.

```
Command Prompt - mysql -u latif -p
mysql> CREATE TABLE karyawan(
-> id INTEGER(4),
-> nama VARCHAR(20),
-> jabatan VARCHAR(20),
-> PRIMARY KEY(id),
-> UNIQUE KEY(jabatan));
Query OK, 0 rows affected (0.22 sec)

mysql>
```

Gambar 3.16 Penggunaan unique key.

Terlihat pada gambar di atas pembuatan unique key terdapat pada baris terakhir dan pada kolom jabatan. Unique key dapat lebih dari satu dan dapat dilakukan secara bersamaan.

```
Command Prompt - mysql -u latif -p
mysql> CREATE TABLE biodata_diri(
-> id INTEGER(4),
-> nama VARCHAR(20),
-> alamat VARCHAR(20),
-> no_telpon INTEGER(13),
-> jabatan VARCHAR(20),
-> email VARCHAR(20),
-> PRIMARY KEY(id),
-> UNIQUE KEY(nama,no_telpon));
Query OK, 0 rows affected (0.14 sec)
```

Gambar 3.17 Pembuatan unique key lebih dari satu.

Unique key pada tabel di atas terdapat pada kolom/field nama dan no_telpon. Kolom di atas dipilih sebagai unique key karena dianggap sebagai kolom yang unik. Pada kolom-kolom kemungkinan untuk mempunyai data yang sama tidak mungkin. Setelah di-describe, akan tampak seperti berikut.

```
Command Prompt - mysql -u latif -p
mysql> desc biodata_diri;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(4)        | NO   | PRI |          |       |
| nama  | varchar(20)   | YES  | MUL |          |       |
| alamat | varchar(20)   | YES  |     |          |       |
| no_telpon | int(13)      | YES  |     |          |       |
| jabatan | varchar(20)   | YES  |     |          |       |
| email  | varchar(20)   | YES  |     |          |       |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

Gambar 3.18 Spesifikasi tabel biodata_diri dengan unique key lebih dari satu.

FOREIGN KEY

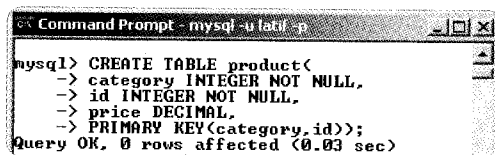
Foreign key merupakan batasan/constrain untuk membuang ketidak-konsistenan dua tabel ketika data pada sebuah tabel tergantung kepada data yang ada di tabel lain. Sintaks foreign key sebagai berikut:

```
[CONSTRAINT constraint_name FOREIGN KEY (colom_
name

[,colom_name [, ...]] )

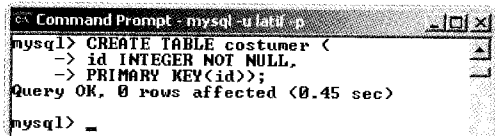
REFERENCES table_name (kolom_name [,kolom_name
[,...]] ) ]
```

Untuk memperjelas sintaks di atas lihat contoh berikut.



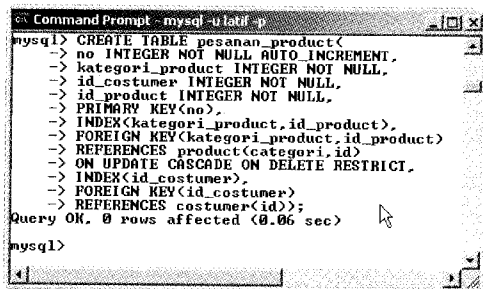
Gambar 3.19 Tabel product.

Setelah Anda membuat tabel product, kemudian buat table costumer.



Gambar 3.20 Tabel costumer.

Setelah Anda membuat tabel costumer, buatlah tabel **pesanan_product**. Pada tabel pesanan_product inilah akan diterapkan foreign key, yang menghubungkan table product dan tabel costumer.



Gambar 3.21 Pembuatan foreign key.

MENAMBAH PRIMARY KEY

Apabila Anda belum menentukan kolom mana yang menjadi primary key, Anda dapat menentukannya setelah tabel sudah Anda buat dengan perintah ALTER.

```
ALTER [IGNORE] TABLE table_name

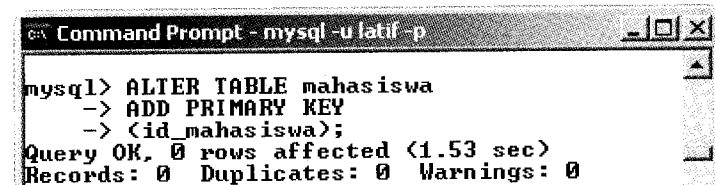
ADD [CONSTRAINT [symbol]]

PRIMARY KEY [index_type] (index_col_
name, ....
```

Field	Type	Null	Key	Default	Extra
id_mahasiswa	int(4)	YES		NULL	
nama	varchar(20)	YES		NULL	
alamat	varchar(20)	YES		NULL	

Gambar 3.22 Spesifikasi tabel mahasiswa.

Tabel di atas adalah table yang belum terdapat PRIMARY KEY. Anda dapat menambahkan PRIMARY KEY dengan perintah ADD PRIMARY KEY.



Gambar 3.23 Menambah primary key.

Setelah Anda menjalankan perintah di atas, tabel mahasiswa akan terdapat primary key. Untuk melihatnya, Anda dapat menggunakan desc yang telah dijelaskan di atas. Spesifikasi tabel mahasiswanya sebagai berikut.

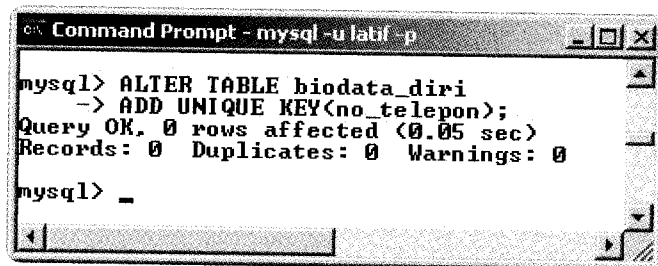
Field	Type	Null	Key	Default	Extra
id_mahasiswa	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
alamat	varchar(20)	YES		NULL	

Gambar 3.24 Table mahasiswa setelah ditambah primary key.

MENAMBAH UNIQUE KEY

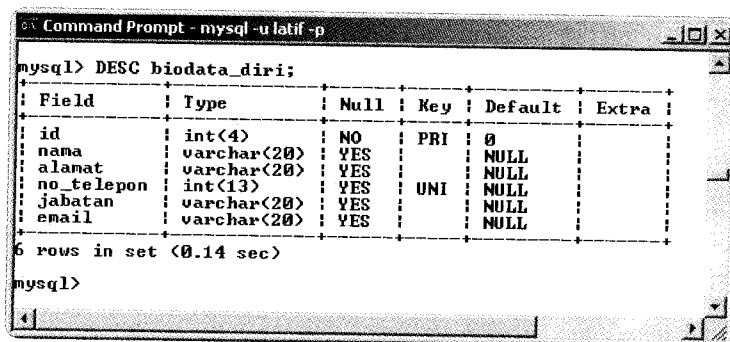
Sama seperti menambah primary key, unique key pun dapat ditambahkan pada tabel yang belum terdapat unique key dan bahkan dapat ditambahkan walaupun sudah ada unique key, karena unique key dapat lebih dari satu.

```
ALTER [IGNORE] TABLE table_name  
  
ADD [CONSTRAINT [symbol]]  
  
UNIQUE [INDEX | KEY] [index_name] [index_type]  
  
(index_colm_name,...)
```



Gambar 3.25 Menambah unique key.

Setelah Anda menjalankan perintah di atas, unique key akan ditambahkan pada tabel `biodata_diri`. Spesifikasi tabel `biodata_diri` menjadi:



Gambar 3.26 Penambahan unique key pada tabel `biodata_diri`.

Terlihat pada gambar di atas kolom key ada kata UNI, ini berarti kolom `no_telepon` telah menjadi unique key.

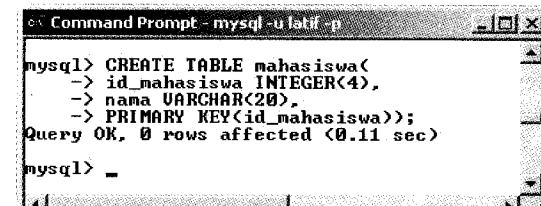
MENAMBAH FOREIGN KEY

Menambah foreign key tidak jauh beda dengan primary key dan unique key. Hanya saja untuk penambahan foreign key melibatkan 2 tabel atau lebih. Perhatikan sintaks berikut:

```
ALTER [IGNORE] TABLE table_name  
  
ADD [CONSTRAINT [symbol]]  
  
FOREIGN KEY [index_name] (index_col_name,...)
```

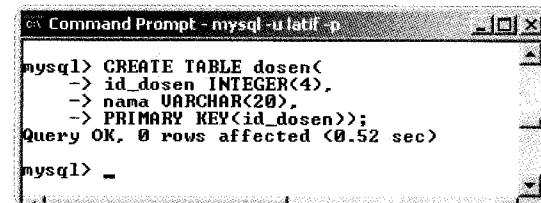
Untuk lebih jelasnya, perhatikan contoh berikut:

Buatlah tabel dengan nama mahasiswa seperti di bawah ini.



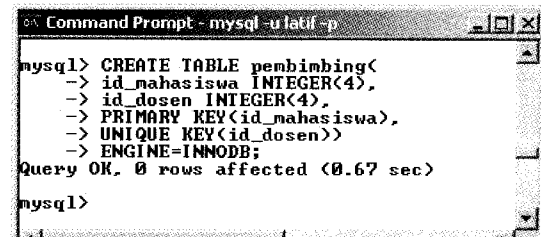
Gambar 3.27 Membuat tabel mahasiswa.

Setelah Anda membuat tabel mahasiswa, sekarang buatlah tabel dosen.



Gambar 3.28 Membuat tabel dosen.

Setelah Anda membuat tabel mahasiswa dan dosen, sekarang buatlah tabel pembimbing.



Gambar 3.29 Membuat table pembimbing.

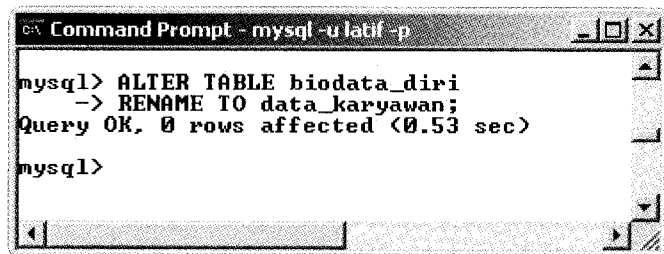
Setelah Anda membuat semua tabel, Anda dapat menambahkan foreign key. Terlihat pada tabel pembimbing di atas yang mampu menghubungkan antara tabel mahasiswa dan table dosen, tetapi tabel pembimbing belum dibuat foreign key agar kedua tabel itu terhubung.

MENGUBAH NAMA TABLE

Kadang, saat Anda memberi nama pada tabel tidak sesuai dengan yang Anda inginkan atau rancu. Untuk mengubah nama tersebut, gunakan perintah berikut:

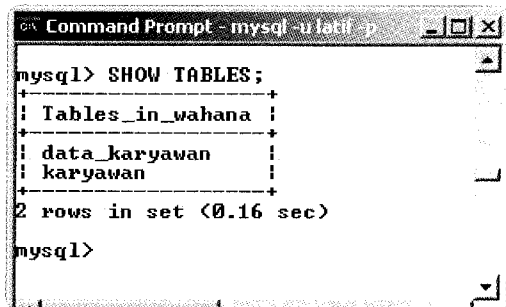
```
ALTER [IGNORE] TABLE table_name
```

```
RENAME [TO] new_table_name
```



Gambar 3.30 Mengubah nama tabel.

Untuk melihat apakah nama tabel biodata_diri sudah berubah namanya menjadi data_karyawan, gunakan perintah SHOW.



Gambar 3.31 Melihat isi database.

Terlihat pada gambar di atas tabel biodata_diri namanya sudah berubah menjadi data_karyawan.

MENAMBAH KOLOM/FIELD

Saat Anda membuat tabel terkadang ada data yang belum dimasukkan ke dalam kolom dan ternyata data tersebut belum memiliki kolom tersendiri. Untuk solusinya, Anda dapat menambahkan kolom pada tabel tersebut. Untuk menambahkan kolom, gunakan perintah berikut:

```
ALTER [IGNORE] TABLE table_name
```

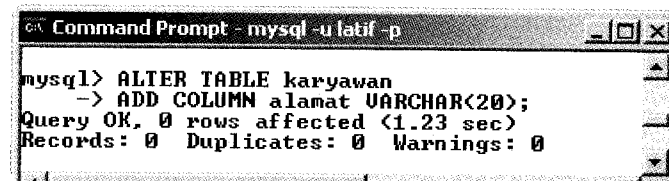
```
ADD [COLUMN] (column_definition,...)
```

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	YES	UNI	NULL	

3 rows in set (0.02 sec)

Gambar 3.32 Tampilan tabel karyawan dengan tiga kolom.

Pada tabel karyawan terdapat tiga kolom/field tentunya kurang lengkap untuk informasi biodata karyawan. Dengan perintah ALTER, Anda dapat menambahkan kolom untuk menambah informasi karyawan. Sebagai contoh, Anda menambahkan alamat karyawan.



Gambar 3.33 Perintah penambahan kolom alamat pada tabel karyawan.

Setelah Anda menjalankan perintah di atas, tabel karyawan akan bertambah kolom alamat.

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	YES	UNI	NULL	
alamat	varchar(20)	YES		NULL	

4 rows in set (0.17 sec)

Gambar 3.34 Tabel karyawan setelah ditambah satu kolom alamat.

MENGURANGI KOLOM

Di samping menambah kolom pada tabel, Anda juga dapat mengurangi atau menghapusnya. Anda dapat melakukan perintah ini apabila Anda kelebihan kolom pada suatu tabel atau terdapat suatu kolom yang tidak berguna.

```
ALTER [IGNORE] TABLE table_name
```

```
DROP [COLUMN] col_name
```

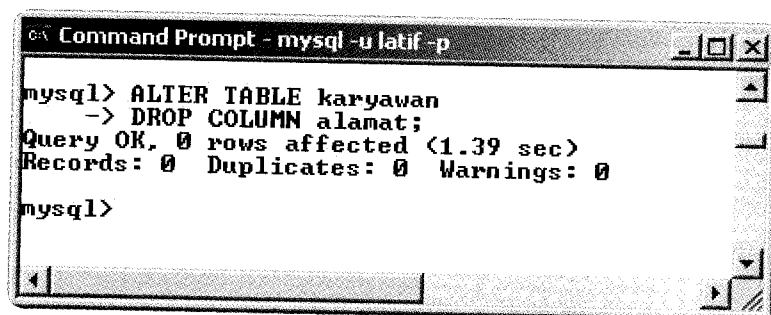
Berikut adalah kolom-kolom pada tabel karyawan.

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	YES	UNI	NULL	
alamat	varchar(20)	YES		NULL	

4 rows in set (0.13 sec)

Gambar 3.35 Spesifikasi pada tabel mahasiswa.

Gambar di atas adalah spesifikasi pada tabel mahasiswa. Terlihat di atas terdapat empat kolom, yaitu id, nama, jabatan, alamat. Sebagai contoh, kolom alamat sudah ada pada tabel lain, sehingga Anda harus menghilangkannya agar tidak ada data yang ganda. Caranya, jalankan perintah berikut:



Gambar 3.36 Perintah menghapus kolom.

Setelah Anda menjalankan perintah di atas, kolom pada tabel karyawan akan berkurang satu, yaitu kolom alamat.

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	YES	UNI	NULL	

3 rows in set (1.00 sec)

Gambar 3.37 Spesifikasi tabel karyawan setelah di drop.

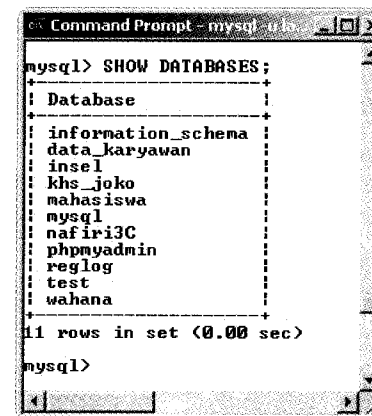
Pada gambar di atas terlihat tabel karyawan hanya terdapat 3 kolom, yaitu id, nama, jabatan. Kolom alamat sudah tidak ada karena sudah dihapus dari tabel karyawan.

MENGHAPUS DATABASE

Apabila Anda sudah tidak memerlukan database yang telah Anda buat, Anda dapat menghapusnya. Apabila Anda menghapus database berarti tabel-tabel di dalamnya akan ikut terhapus pula. Hati-hati dengan perintah ini, apabila Anda sudah menjalankan perintah ini, database yang telah Anda hapus tidak dapat dikembalikan lagi. Kecuali Anda telah membackup database tersebut. Berikut sintaksnya:

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name;
```

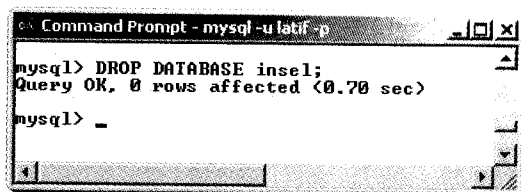
Berikut adalah beberapa database yang terdapat dalam komputer ini.



Gambar 3.38 Menampilkan database.

Di atas adalah beberapa database yang terdapat dalam komputer

penulis. Sebagai contoh, saya ingin menghapus database insel dengan menggunakan statement ini.

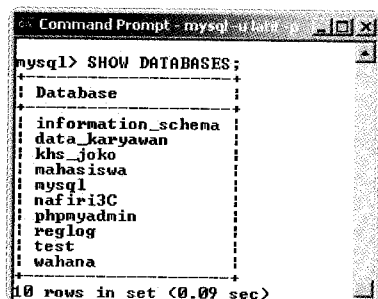


```
mysql> DROP DATABASE insel;
Query OK, 0 rows affected (0.70 sec)

mysql> _
```

Gambar 3.39 Perintah menghapus database.

Setelah perintah di atas dijalankan, database insel akan terhapus. Untuk melihat apakah sudah terhapus atau belum, dapat menggunakan perintah SHOW.



```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| data_karyawan      |
| khs_joko           |
| mahasiswa          |
| mysql              |
| nafiri3C           |
| phpmyadmin         |
| reglog             |
| test               |
| wahana             |
+-----+
10 rows in set (0.09 sec)
```

Gambar 3.40 Menampilkan database.

Terlihat pada gambar di atas database insel sudah tidak ada, ini berarti database insel telah terhapus.

MENGHAPUS TABEL

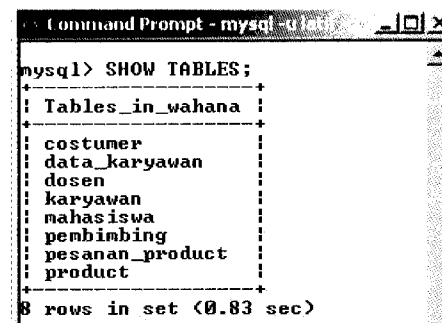
Anda dapat menghapus satu atau lebih tabel. Untuk menghapusnya, Anda harus mempunyai hak akses DROP untuk masing-masing tabel. Apabila Anda menjalankan perintah ini, semua data pada tabel akan ikut terhapus pula. Oleh karena itu, hati-hati dengan statement ini.

```
DROP [TEMPORARY] TABLE [IF EXISTS]
```

```
Table_name [,table_name] ...
```

```
[RESTRICT | CASCADE]
```

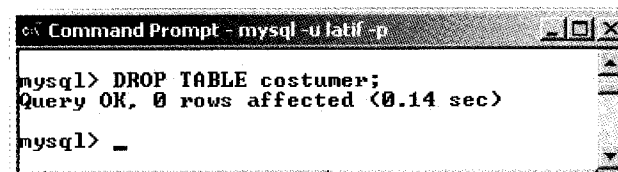
Pertama, untuk menghapus tabel, lihat nama tabel tersebut. Untuk melihat tabel gunakan statement SHOW.



```
mysql> SHOW TABLES;
+-----+
| Tables_in_wahana |
+-----+
| costumer          |
| data_karyawan     |
| dosen             |
| karyawan          |
| mahasiswa         |
| pembimbing        |
| pesanan_product   |
| product           |
+-----+
8 rows in set (0.83 sec)
```

Gambar 3.41 Menampilkan semua tabel.

Gambar di atas adalah tabel-tabel yang terdapat pada database wahana. Anda dapat menghapus tabel-tabel tersebut dengan statement ini. Sebagai contoh, saya ingin menghapus tabel costumer.

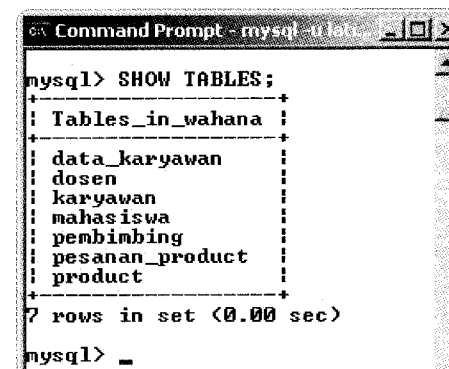


```
mysql> DROP TABLE costumer;
Query OK, 0 rows affected (0.14 sec)

mysql> _
```

Gambar 3.42 Menghapus tabel costumer.

Setelah menjalankan perintah tersebut, tabel costumer akan terhapus dari database wahana. Untuk melihat apakah tabel costumer tersebut sudah benar-benar terhapus, gunakan SHOW.



```
mysql> SHOW TABLES;
+-----+
| Tables_in_wahana |
+-----+
| data_karyawan     |
| dosen             |
| karyawan          |
| mahasiswa         |
| pembimbing        |
| pesanan_product   |
| product           |
+-----+
7 rows in set (0.00 sec)

mysql> _
```

Gambar 3.43 Menampilkan tabel pada database wahana.

Pada gambar di atas, table costumer sudah tidak ada dalam database wahana.

MENGHAPUS PRIMARY KEY

Anda dapat menghapus primary key yang telah Anda tetapkan sebelumnya. Apabila dalam suatu kejadian kolom yang dijadikan sebagai primary key ternyata banyak data yang sama. Apabila kolom tersebut dipertahankan menjadi primary key, data selanjutnya tidak akan bisa dimasukkan. Berikut sintaksnya:

```
ALTER [IGNORE] TABLE table_name
```

```
DROP PRIMARY KEY
```

Sebagai contoh, di bawah ini adalah table karyawan.

Field	Type	Null	Key	Default	Extra
id	int(4)	NO		0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	NO	PRI		

Gambar 3.44 Spesifikasi tabel karyawan.

Pada gambar di atas terlihat jabatan sebagai primary key. Karena jabatan kemungkinan ada yang sama, Anda dapat menghapusnya dengan statement ini.

```
mysql> ALTER TABLE karyawan
-> DROP PRIMARY KEY;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql>
```

Gambar 3.45 Statement untuk menghapus primary key.

Setelah menjalankan perintah di atas, primary key yang terdapat pada tabel jabatan akan terhapus. Sekarang pada tabel karyawan tidak terdapat primary key. Anda dapat menggantinya dengan kolom lain yang dimungkinkan tidak ada data yang sama. Pada tabel karyawan yang kemungkinan tidak ada data yang sama adalah kolom id. Dengan demikian, id dapat ditetapkan sebagai primary key. Untuk menetapkan kolom id sebagai primary key, Anda dapat menggunakan perintah ADD.

```
mysql> ALTER TABLE karyawan
-> ADD PRIMARY KEY(id);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql>
```

Gambar 3.46 Penambahan primary key pada kolom id.

Setelah menjalankan perintah di atas, kolom id akan menjadi primary key.

```
mysql> DESC karyawan;
```

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	NO			

3 rows in set (0.05 sec)

Gambar 3.47 Kolom id menjadi primary key.

MENGHAPUS UNIQUE KEY

Menghapus unique key hampir sama dengan menghapus primary key. Perbedaannya hanya pada suatu tabel unique key bisa satu ataupun lebih. Berikut sintaknya:

```
ALTER [IGNORE] TABLE table_name
```

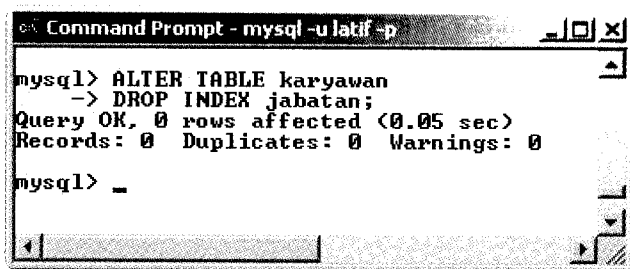
```
DROP {INDEX | KEY} index_name
```

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	NO	UNI		

Gambar 3.48 Gambar spesifikasi table karyawan.

Pada gambar di atas, terlihat kolom jabatan sebagai unique key. Apabila ada jabatan yang sama, data yang terakhir dimasukkan tidak akan bisa.

Jadi, apabila Anda ingin menghapus unique key, gunakan statement ini.



```
Command Prompt - mysql -u latif -p

mysql> ALTER TABLE karyawan
-> DROP INDEX jabatan;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> _
```

Gambar 3.49 Statement penghapusan unique key.

Setelah Anda menjalankan perintah di atas, kolom jabatan sudah tidak menjadi unique key lagi. Spesifikasi tabel karyawan menjadi berikut.

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	0	
nama	varchar(20)	YES		NULL	
jabatan	varchar(20)	NO			

Gambar 3.50 Spesifikasi tabel karyawan setelah didrop.

INDEX

Ketika data dalam tabel bertambah besar, proses pembacaan data akan menjadi lambat. Disinilah penggunaan index akan terlihat perannya karena memungkinkan pencarian data menjadi lebih cepat. Selain mempercepat proses pencarian, dengan adanya index Anda dapat melakukan pengunikan (unique) untuk mencegah duplikasi data.

Pada MySQL mendukung empat tipe index:

- Non-unique simple indexes
- UNIQUE index, setiap nilai adalah berbeda dari yang lainnya
- PRIMARY KEY adalah UNIQUE key yang tidak membolehkan nilai NULL.
- FULLTEXT dioptimasi untuk pencarian teks.

Index dapat didefinisikan ketika tabel dibuat. Ini meliputi definisi index dalam perintah CREATETABLE, bersama dengan pendefinisian kolom.

Suatu definisi index terdiri dari suatu kata kunci yang menunjukkan tipe index, diikuti oleh suatu daftar nama-nama kolom yang akan dijadikan index di dalam tanda kurung. Semua itu sudah dijelaskan.

QUERY

MySQL mendukung penuh ANSI SQL-92. Referensi SQL untuk MySQL adalah referensi SQL yang umum digunakan secara luas. SQL adalah bahasa query database yang serupa dengan bahasa Inggris yang terdiri dari susunan kata kerja. Setiap susunan kata kerja ini, diawali dengan sebuah perintah SQL, diikuti oleh keyword, literals, identifiers, atau punctuation.

Dalam bagian ini menggambarkan prinsip-prinsip dasar penulisan perintah-perintah SQL, bekerja dengan berbagai query, dari dasar hingga yang kompleks. Query menggambarkan beberapa hal tentang klien mysql:

- Ketika Anda menuliskan dan mengeksekusi perintah, program klien mysql mengirimkannya ke server untuk dieksekusi dan menampilkan hasilnya, kemudian mencetak prompt mysql> kembali untuk menunjukkan bahwa klien mysql sudah siap menerima perintah lainnya.
- Klien mysql menampilkan output dari query dalam bentuk tabular (terdiri dari baris dan kolom). Baris pertama menunjukkan nama-nama kolom tabel dan baris-baris berikutnya adalah hasil query.
- Klien mysql menampilkan juga informasi mengenai berapa banyak baris yang dikembalikan dan berapa lama query tersebut dieksekusi, yang memberikan Anda ide pemikiran tentang performa server.

Query MySQL garis besarnya terdiri dari statement-statement, di antaranya statement SELECT, INSERT, REPLACE, UPDATE, SELECT, dan DELETE.

STATEMENT SELECT

Umumnya, sebuah data disimpan ke dalam database, karena nantinya data tersebut ingin ditampilkan kembali. Cara menampilkan meng-

gunakan perintah select. Syntax dasar untuk perintah select di MySQL adalah:

```
SELECT daftar_pilihan_kolom FROM daftar_tabel

WHERE batasan_batasan_utama

GROUP BY grup_kolom

ORDER BY urutan_kolom

HAVING batasan_batasan_tambahan

LIMIT jumlah
```

Agar Anda dapat memahami sintaksnya, berikut akan diuraikan satu per satu:

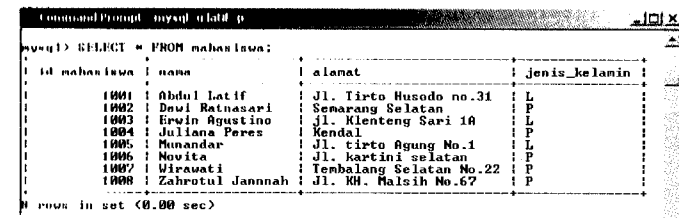
- `daftar_pilihan_kolom`, menunjukkan kolom-kolom apa saja yang akan dipilih atau ditampilkan.
- `daftar_tabel`, menunjukkan dari tabel apa saja baris-baris data diambil atau dipilih.
- `batasan_batasan_utama`, menunjukkan apa saja kondisi-kondisi baris data yang harus terpenuhi.
- `grup_kolom`, menunjukkan bagaimana mengelompokkan hasil-hasil query.
- `urutan_kolom`, menunjukkan bagaimana mengurutkan hasil-hasil query.
- `batasan_batasan_tambahan`, menunjukkan apa saja kondisi-kondisi tambahan dari baris data yang harus terpenuhi.
- `jumlah`, menunjukkan batas dari hasil query.

Pengambilan data dapat dilakukan dengan beberapa cara di antaranya adalah:

Pengambilan data sederhana

Perintah SELECT sederhana untuk pengambilan data dari tabel dapat dilakukan tanpa harus mengisi atribut-atribut yang banyak. Dapat

Anda lihat pada contoh berikut.



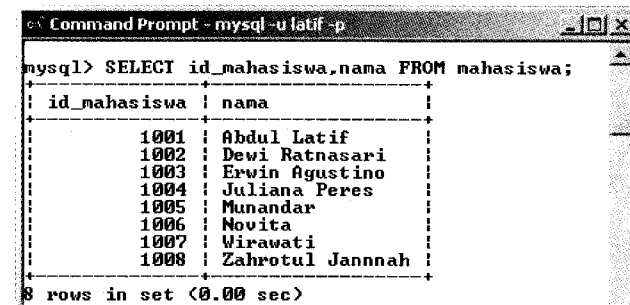
```
mysql> SELECT * FROM mahasiswa;
```

id_mahasiswa	nama	alamat	jenis_kelamin
1001	Abdul Latif	Jl. Tirta Husodo no.31	L
1002	Dewi Ratnasari	Semarang Selatan	P
1003	Erwin Agustino	Jl. Klenteng Sari 1A	L
1004	Juliana Peres	Kendal	P
1005	Munandar	Jl. tirta Agung No.1	L
1006	Novita	Jl. kartini selatan	P
1007	Wirawati	Tembalang Selatan No. 22	P
1008	Zahrotul Jannah	Jl. KH. Malsih No.67	P

8 rows in set (0.00 sec)

Gambar 3.51 Pengambilan data sederhana.

Anda juga dapat mengambil data berdasarkan kolom-kolom. Caranya dengan menentukan nama-nama kolom yang akan diambil, sebagai berikut.



```
mysql> SELECT id_mahasiswa, nama FROM mahasiswa;
```

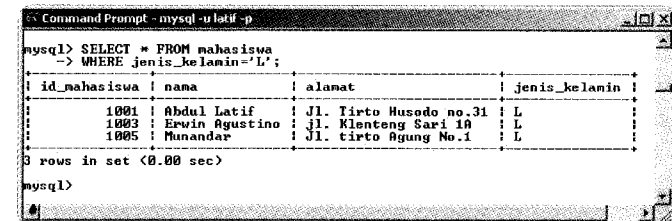
id_mahasiswa	nama
1001	Abdul Latif
1002	Dewi Ratnasari
1003	Erwin Agustino
1004	Juliana Peres
1005	Munandar
1006	Novita
1007	Wirawati
1008	Zahrotul Jannah

8 rows in set (0.00 sec)

Gambar 3.52 Pengambilan data berdasarkan kolom.

Membatasi suatu perintah SELECT dengan WHERE

Dengan menggunakan WHERE, Anda dapat mengambil atau memilih baris-baris data tertentu saja dari tabel Anda. Berikut contohnya.



```
mysql> SELECT * FROM mahasiswa
-> WHERE jenis_kelamin='L';
```

id_mahasiswa	nama	alamat	jenis_kelamin
1001	Abdul Latif	Jl. Tirta Husodo no.31	L
1003	Erwin Agustino	Jl. Klenteng Sari 1A	L
1005	Munandar	Jl. tirta Agung No.1	L

3 rows in set (0.00 sec)

Gambar 3.53 Seleksi dengan operator sama dengan (=).

Gambar di atas membatasi suatu perintah select dengan where, dengan menggunakan operator sama dengan (=). Selain menampilkan semua kolom, Anda juga dapat menampilkannya berdasarkan kolom.

```

Command Prompt - mysql -u latif -p
mysql> SELECT nama FROM mahasiswa
-> WHERE jenis_kelamin='L';
+-----+
| nama |
+-----+
| Abdul Latif |
| Erwin Agustino |
| Munandar |
+-----+
3 rows in set (0.03 sec)

mysql> _

```

Gambar 3.54 Menyeleksi dengan where berdasarkan kolom.

Gambar di atas adalah tampilan data pada nama yang berjenis kelamin laki-laki (L). Semua laki-laki akan ditampilkan bila Anda menjalankan perintah di atas. Jika Anda menginginkan sebaliknya, Anda dapat menggunakan operator tidak sama dengan (!=). Berikut contohnya.

```

Command Prompt - mysql -u latif -p
mysql> SELECT nama FROM mahasiswa
-> WHERE jenis_kelamin != 'L';
+-----+
| nama |
+-----+
| Dewi Ratnasari |
| Juliana Peres |
| Novita |
| Wirawati |
| Zahrotul Jannnah |
+-----+
5 rows in set (0.01 sec)

mysql> _

```

Gambar 3.55 Seleksi data menggunakan operator tidak sama dengan(!=).

Pada gambar di atas sintaks untuk menampilkan yang bukan laki-laki (L). Jadi, data yang ditampilkan adalah berjenis kelamin perempuan (P) karena pada kasus ini menggunakan operator tidak sama dengan (!=). Selain itu, Anda juga menampilkan data pada sebuah tabel yang melebihi data yang ditentukan operator. Berikut contohnya.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
-> WHERE id_mahasiswa > '1003';
+-----+-----+-----+-----+
| id_mahasiswa | nama | alamat | jenis_kelamin |
+-----+-----+-----+-----+
| 1004 | Juliana Peres | Kendal | P |
| 1005 | Munandar | Jl. Tirta Agung No.1 | L |
| 1006 | Novita | Jl. Kartini Selatan | P |
| 1007 | Wirawati | Tembalang Selatan No.22 | P |
| 1008 | Zahrotul Jannnah | Jl. KH. Malsih No.67 | P |
+-----+-----+-----+-----+
5 rows in set (0.13 sec)

mysql>

```

Gambar 3.56 Menampilkan data dengan operator ">".

Data yang ditampilkan di atas adalah data yang nilainya lebih dari 1003. Apabila Anda ingin menampilkan data pada suatu tabel yang nilainya kurang dari nilai pembandingan, Anda dapat menggunakan operator kurang dari (<).

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
-> WHERE id_mahasiswa < '1005';
+-----+-----+-----+-----+
| id_mahasiswa | nama | alamat | jenis_kelamin |
+-----+-----+-----+-----+
| 1001 | Abdul Latif | Jl. Tirta Husodo no.31 | L |
| 1002 | Dewi Ratnasari | Semarang Selatan | P |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A | L |
| 1004 | Juliana Peres | Kendal | P |
+-----+-----+-----+-----+
4 rows in set (0.19 sec)

mysql>

```

Gambar 3.57 Menampilkan data dengan operator "<".

Karena Anda menjalankan perintah kurang dari (<), yang ditampilkan adalah id_mahasiswa yang kurang dari 1005. Data 1005 tidak ditampilkan karena menggunakan kurang dari (<). Apabila Anda menginginkan data 1005 ditampilkan, gunakan kurang dari sama dengan (<=). Berikut contohnya.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
-> WHERE id_mahasiswa <= '1005';
+-----+-----+-----+-----+
| id_mahasiswa | nama | alamat | jenis_kelamin |
+-----+-----+-----+-----+
| 1001 | Abdul Latif | Jl. Tirta Husodo no.31 | L |
| 1002 | Dewi Ratnasari | Semarang Selatan | P |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A | L |
| 1004 | Juliana Peres | Kendal | P |
| 1005 | Munandar | Jl. Tirta Agung No.1 | L |
+-----+-----+-----+-----+
5 rows in set (0.11 sec)

mysql> _

```

Gambar 3.58 Menampilkan data dengan operator "<=".

Gambar di atas terlihat data 1005 tampak karena menggunakan operator kurang dari sama dengan (<=). Begitu pun sebaliknya, apabila Anda menggunakan operator lebih dari atau sama dengan (>=).

Mengurutkan pemilihan data menggunakan ORDER BY

Dalam contoh sebelumnya, hasil baris-baris data yang ditampilkan tidak dalam urutan tertentu. Biasanya lebih mudah untuk menguji output dari query jika baris-baris data diurutkan. Untuk mengurutkan baris data, gunakan klausa ORDER BY, seperti berikut ini.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
  > WHERE id_mahasiswa ORDER BY nama;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1001 | Abdul Latif  | Jl. Tirta Husodo no.31 | L              |
| 1002 | Dewi Ratnasari | Senarang Selatan      | P              |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A  | L              |
| 1004 | Juliana Peres  | Kendal                | P              |
| 1005 | Munandar      | Jl. tirta Agung No.1  | L              |
| 1010 | Nasrul Kurniawan | Jl. Prapatan mayong no.2A | L              |
| 1006 | Novita        | Jl. kartini selatan   | P              |
| 1009 | Uira Andriani | Jl. Raya Margoyoso no.1 | P              |
| 1007 | Wirawati      | Tembalang Selatan No.22 | P              |
| 1008 | Zahrotul Jannah | Jl. KH. Malsih No.67 | P              |
+----+-----+-----+-----+
10 rows in set (0.09 sec)

mysql> _

```

Gambar 3.59 Menampilkan data menggunakan ORDER BY.

Selain mengurutkan dari yang terkecil ke yang terbesar, Anda juga dapat mengurutkan data dari yang terbesar ke yang terkecil.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
  > id_mahasiswa ORDER BY nama DESC;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1008 | Zahrotul Jannah | Jl. KH. Malsih No.67 | P              |
| 1007 | Wirawati        | Tembalang Selatan No.22 | P              |
| 1009 | Uira Andriani   | Jl. Raya Margoyoso no.1 | P              |
| 1006 | Novita          | Jl. kartini selatan   | P              |
| 1010 | Nasrul Kurniawan | Jl. Prapatan mayong no.2A | L              |
| 1005 | Munandar        | Jl. tirta Agung No.1  | L              |
| 1004 | Juliana Peres   | Kendal                | P              |
| 1003 | Erwin Agustino  | Jl. Klenteng Sari 1A  | L              |
| 1002 | Dewi Ratnasari  | Senarang Selatan      | P              |
| 1001 | Abdul Latif     | Jl. Tirta Husodo no.31 | L              |
+----+-----+-----+-----+
10 rows in set (3.16 sec)

```

Gambar 3.60 Menampilkan data dengan teknik Descending.

Terlihat pada kolom nama datanya dari aljabar yang paling terakhir berurut ke aljabar yang paling awal, ini karena teknik descending.

Membatasi pengambilan data menggunakan klausa LIMIT

Klausa LIMIT memungkinkan Anda untuk membatasi output dari query. Klausa LIMIT sangat berguna, khususnya bersama dengan klausa ORDER

BY. Ini berguna untuk mengambil baris baris data berdasarkan posisinya dalam sekumpulan baris data hasil pemilihan. LIMIT mungkin diberikan dengan satu atau dua argument, yang mana berupa bilangan bulat.

LIMIT jumlah_baris

LIMIT jumlah_diabaikan, jumlah_baris

Pada sintaks di atas, jumlah_baris menunjukkan berapa banyak baris yang akan ditampilkan, sedangkan jumlah_diabaikan menunjukkan berapa banyak baris data yang pertama dari hasil pengambilan data yang harus diabaikan untuk tidak ditampilkan. Berikut ini contoh penggunaan klausa LIMIT untuk menampilkan 3 baris data yang pertama.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
  > WHERE id_mahasiswa LIMIT 3;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1001 | Abdul Latif  | Jl. Tirta Husodo no.31 | L              |
| 1002 | Dewi Ratnasari | Senarang Selatan      | P              |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A  | L              |
+----+-----+-----+-----+
3 rows in set (1.55 sec)

```

Gambar 3.61 Menampilkan data dengan batasan 3 baris.

Jika LIMIT diikuti oleh dua bilangan bulat, jumlah_diabaikan dan jumlah_baris, LIMIT akan mengabaikan sejumlah baris data pertama yang ditetapkan oleh bilangan bulat yang pertama (jumlah_diabaikan), dan hanya akan menampilkan sejumlah baris data berikutnya sebanyak yang ditetapkan oleh bilangan bulat yang kedua (jumlah_baris). Untuk mengabaikan dua baris data yang pertama dan menampilkan dua baris data berikutnya, dapat Anda lihat dalam contoh berikut.

```

Command Prompt - mysql -u latif -p
mysql> SELECT * FROM mahasiswa
  > WHERE id_mahasiswa LIMIT 3,5;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1004 | Juliana Peres  | Kendal                | P              |
| 1005 | Munandar      | Jl. tirta Agung No.1  | L              |
| 1006 | Novita        | Jl. kartini selatan   | P              |
| 1007 | Wirawati      | Tembalang Selatan No.22 | P              |
| 1008 | Zahrotul Jannah | Jl. KH. Malsih No.67 | P              |
+----+-----+-----+-----+
5 rows in set (1.55 sec)

```

Gambar 3.62 Menampilkan data dengan dari data ke 3 sampai ke 5.

Mengelompokkan pemilihan baris data menggunakan klausa GROUP BY

GROUP BY adalah fungsi untuk mengelompokkan data dalam sebuah

kolom yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Jadi, apabila dalam satu kolom terdapat beberapa data yang sama, data yang akan ditampilkan hanya salah satu. Berikut contohnya.

```

mysql> SELECT * FROM mahasiswa;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1001 | Abdul Latif  | Jl. Tirta Husodo no.31 | L              |
| 1002 | Dewi Ratnasari | Semarang Selatan      | P              |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A  | L              |
| 1004 | Juliana Peres  | Kendal                | P              |
| 1005 | Munandar      | Jl. tirta Agung No.1  | L              |
| 1006 | Novita        | Jl. kartini selatan   | P              |
| 1007 | Wirawati      | Tembalang Selatan No.22 | P              |
| 1008 | Zahrotul Jannah | Jl. KH. Malsih No.67 | P              |
| 1009 | Uira Andriani | Jl. Raya Margoyoso no.1 | P              |
| 1010 | Nasrul Kurniawan | Jl. Prapatan mayong no.2A | L              |
+----+-----+-----+-----+
10 rows in set (0.36 sec)

mysql> _

```

Gambar 3.63 Menampilkan data pada tabel mahasiswa.

Gambar di atas adalah semua data yang terdapat tabel mahasiswa. Sekarang, Anda akan memberi contoh fungsi GROUP BY pada tabel mahasiswa.

```

mysql> SELECT id_mahasiswa,nama,jenis_kelamin
-> FROM mahasiswa GROUP BY jenis_kelamin;
+----+-----+-----+
| id_mahasiswa | nama          | jenis_kelamin |
+----+-----+-----+
| 1001 | Abdul Latif  | L              |
| 1002 | Dewi Ratnasari | P              |
+----+-----+-----+
2 rows in set (0.17 sec)

mysql> _

```

Gambar 3.64 Menampilkan data dengan klausa/fungsi group by.

PERINTAH REPLACE

Perintah REPLACE mengembalikan sebuah informasi yang menunjukkan jumlah baris yang dipengaruhi. Jika jumlah baris yang dipengaruhi oleh perintah REPLACE adalah satu, baris data hanya ditambahkan ke dalam

tabel tanpa ada proses mengganti baris data yang ada. Jika jumlah baris data yang dipengaruhi oleh perintah REPLACE adalah dua baris, maka sebuah baris yang dihapus sebelum proses penambahan baris data dilakukan. Perintah REPLACE memiliki format penulisan perintah yang sama dengan INSERT, yaitu sebagai berikut:

```

REPLACE [LOW_PRIORITY | DELAYED]

[INTO] table_name [(col_name,...)]

VALUES ({ expr | DEFAULT},...), (...),...

```

Atau

```

REPLACE INTO nama_tabel (daftar_kolom)

VALUES (daftar_nilai);

```

Agar Anda lebih memahami statement di atas, berikut contohnya.

```

mysql> SELECT * FROM mahasiswa;
+----+-----+-----+-----+
| id_mahasiswa | nama          | alamat                | jenis_kelamin |
+----+-----+-----+-----+
| 1001 | Abdul Latif  | Jl. Tirta Husodo no.31 | L              |
| 1002 | Dewi Ratnasari | Semarang Selatan      | P              |
| 1003 | Erwin Agustino | Jl. Klenteng Sari 1A  | L              |
| 1004 | Juliana Peres  | Kendal                | P              |
| 1005 | Munandar      | Jl. tirta Agung No.1  | L              |
| 1006 | Novita        | Jl. kartini selatan   | P              |
| 1007 | Wirawati      | Tembalang Selatan No.22 | P              |
| 1008 | Zahrotul Jannah | Jl. KH. Malsih No.67 | P              |
| 1009 | Uira Andriani | Jl. Raya Margoyoso no.1 | P              |
| 1010 | Nasrul Kurniawan | mayong                | L              |
+----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> _

```

Gambar 3.65 Menampilkan data pada table mahasiswa.

Gambar di atas adalah data pada tabel mahasiswa. Sebagai contoh, Anda me-replace data dengan id_mahasiswa 1001.

```

mysql> REPLACE INTO mahasiswa VALUES
-> ('1001','Latif','jepara','L');
Query OK, 2 rows affected (0.17 sec)

mysql> _

```

Gambar 3.66 Perintah me-replace pada table mahasiswa.

Setelah Anda menjalankan perintah di atas, baris pertama akan mengalami perubahan sesuai dengan data yang Anda masukkan.

```
mysql> SELECT * FROM mahasiswa;
```

id_mahasiswa	nama	alamat	jenis_kelamin
1001	Latif	Jepara	L
1002	Dewi Ratnasari	Semarang Selatan	P
1003	Erwin Agustino	Jl. Klenteng Sari 1A	L
1004	Juliana Peres	Kendal	P
1005	Munandar	Jl. tirta Agung No.1	L
1006	Novita	Jl. Kartini selatan	P
1007	Wirawati	Tembalang Selatan No.22	P
1008	Zahrotul Jannah	Jl. KH. Malsih No.67	P
1009	Uira Andriani	Jl. Raya Margoyoso no.1	P
1010	Nasrul Kurniawan	mayong	L

```
10 rows in set (0.00 sec)

mysql>
```

Gambar 3.67 Menampilkan data setelah di-replace.

Terlihat di atas pada baris pertama terjadi perubahan pada kolom nama dan alamat. Keuntungan menggunakan perintah REPLACE adalah Anda dapat menjalankan perintah DELETE dan INSERT sekaligus dalam sebuah operasi tunggal. Di sini tidak ada kebutuhan untuk melakukan penguncian tabel secara eksplisit sebagaimana yang mungkin ada ketika Anda memutuskan untuk memisahkan perintah DELETE dan INSERT.

MEMPERBARUI DATA (UPDATE)

Command UPDATE dipakai untuk mengubah isi data di baris-baris yang ada. Cara menggunakannya adalah dengan menentukan nama tabel yang akan diubah isi datanya. Bisa juga ditampilkan fungsi SET yang diikuti dengan daftar pasangan nama kolom dan nilainya, dan biasanya perintah UPDATE diikuti dengan fungsi WHERE yang menentukan baris-baris data mana sajakah yang akan diubah. Berikut format penulisannya:

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name

    SET col_name1=expr1 [, col_
name2=expr2 ...]

    [WHERE where_condition]

    [ORDER BY ...]

    [LIMIT row_count]
```

Atau,

UPDATE nama_tabel SET

Kolom1='data_baru',

Kolom2='data_baru',

.....

kolomn='data_baru'

WHERE [kondisi];

Sebagai contoh, Anda coba menggunakan tabel mahasiswa.

```
mysql> SELECT * FROM mahasiswa;
```

id_mahasiswa	nama	alamat	jenis_kelamin
1001	Latif	Jepara	L
1002	Dewi Ratnasari	Semarang Selatan	P
1003	Erwin Agustino	Jl. Klenteng Sari 1A	L
1004	Juliana Peres	Kendal	P
1005	Munandar	Jl. tirta Agung No.1	L
1006	Novita	Jl. Kartini selatan	P
1007	Wirawati	Tembalang Selatan No.22	P
1008	Zahrotul Jannah	Jl. KH. Malsih No.67	P
1009	Uira Andriani	Jl. Raya Margoyoso no.1	P
1010	Nasrul Kurniawan	mayong	L

```
10 rows in set (0.34 sec)

mysql>
```

Gambar 3.68 Isi data pada tabel mahasiswa.

Pada suatu kasus mahasiswa yang bernama Erwin Agustino dengan id 1003 pindah alamat ke weleri. Anda harus meng-update alamat mahasiswa tersebut dengan alamatnya yang baru.

```
mysql> UPDATE mahasiswa SET
-> alamat='weleri'
-> WHERE id_mahasiswa='1003';
Query OK, 1 row affected (0.20 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
```

Gambar 3.69 Perintah update pada tabel mahasiswa.

Setelah Anda jalankan perintah gambar 3.69, mahasiswa yang memiliki id 1003, yaitu Erwin Agustino, data alamatnya akan berubah menjadi weleri.

```
mysql> SELECT * FROM mahasiswa;
```

id_mahasiswa	nama	alamat	jenis_kelamin
1001	Latif	Jepara	L
1002	Dewi Ratnasari	Semarang Selatan	P
1003	Erwin Agustino	Kendal	P
1004	Juliana Peres	Jl. Tirta Agung No.1	L
1005	Munandar	Jl. Kartini Selatan	P
1006	Novita	Tembalang Selatan No.22	P
1007	Wirawati	Jl. KH. Malsih No.67	P
1008	Zahrotul Jannah	Jl. Raya Margoyoso no.1	P
1009	Uira Andriani	Mayong	L
1010	Nasrul Kurniawan		L

```
10 rows in set (0.39 sec)

mysql>
```

Gambar 3.70 Tabel mahasiswa setelah di update.

JOIN TABEL

Kemampuan untuk mengakomodasi join bisa dibilang merupakan fitur utama MySQL. Karena itu, sangat penting bagi Anda untuk bisa membuat sebuah perintah select yang mengandung join.

MySQL adalah database relasional, artinya satu kelompok/record data dapat dibagi ke dalam beberapa tabel. Namun, pemisahan tersebut nantinya harus bisa digabungkan kembali sehingga pemisahannya haruslah secara logis.

Karena itulah ada “hubungan khusus” antara tabel-tabel agar nantinya bisa digabungkan lagi. Misalnya, tabel Siswa akan berkaitan erat dengan tabel Pelajaran dan tabel KRS (kartu rencana studi).

Dengan menyimpan informasi ke dalam tempat yang berbeda, ada keuntungan yang bisa diperoleh, yaitu minimalisasi (atau bisa dikatakan penghindaran) data yang ganda/duplikat.

Untuk membuat itu semua, digunakankanlah join. Join adalah kegiatan penggabungan dua buah tabel di mana penggabungannya bersifat relevan, artinya ada hubungan antara nilai di kolom yang dipilih. Join tabel menghasilkan data, di mana data tersebut hanya ada secara temporer, selama query dijalankan.

Penggunaan join untuk sesuatu yang temporer artinya menciptakan baris-baris data yang lengkap dari suatu database yang membagi data-data yang berhubungan melalui atau ke beberapa tabel terpisah (hasil dari normalisasi).

CROSS-JOIN

Cross join antar dua buah tabel adalah mengambil data dari baris data dalam tabel 1 dan menggabungkannya dengan setiap baris-baris data dari tabel 2. Untuk melakukannya, gunakan format perintah berikut:

```
SELECT * FROM table_name1
```

```
CROSS JOIN table_name2;
```

Sebagai contoh, di bawah ini terdapat 2 tabel yaitu tabel matakuliah dan tabel jadwal.

```
mysql> SELECT * FROM matakuliah;
```

kode_matkul	matakuliah
P001	Listrik Magnet
P002	Fisika Dasar
P003	Basis Data
P004	Mikrokontroler
P005	Matematika
P006	Teknik Pengukuran

```
6 rows in set (0.00 sec)

mysql>
```

Gambar 3.71 Data tabel matakuliah.

```
mysql> SELECT * FROM jadwal;
```

kode_matkul	hari	jam
P001	Senin	07:30:00
P002	Senin	09:00:00
P003	Selasa	08:30:00
P004	Rabu	07:30:00
P005	Kamis	09:00:00
P006	Jum'at	09:00:00

```
6 rows in set (0.05 sec)

mysql>
```

Gambar 3.72 Data tabel jadwal.

Langsung saja Anda mencoba meng-cross-join-kan kedua tabel tersebut dengan perintah di atas.

Command Prompt - mysql -u latif -p

```
mysql> SELECT * FROM matakuliah CROSS JOIN jadwal;
```

kode_matkul	matakuliah	kode_matkul	hari	jam
P001	Listrik Magnet	P001	Senin	07:30:00
P002	Fisika Dasar	P001	Senin	07:30:00
P003	Basis Data	P001	Senin	07:30:00
P004	Mikrokontroler	P001	Senin	07:30:00
P005	Matematika	P001	Senin	07:30:00
P006	Teknik Pengukuran	P001	Senin	07:30:00
P001	Listrik Magnet	P002	Senin	09:00:00
P002	Fisika Dasar	P002	Senin	09:00:00
P003	Basis Data	P002	Senin	09:00:00
P004	Mikrokontroler	P002	Senin	09:00:00
P005	Matematika	P002	Senin	09:00:00
P006	Teknik Pengukuran	P002	Senin	09:00:00
P001	Listrik Magnet	P003	Selasa	08:30:00
P002	Fisika Dasar	P003	Selasa	08:30:00
P003	Basis Data	P003	Selasa	08:30:00
P004	Mikrokontroler	P003	Selasa	08:30:00
P005	Matematika	P003	Selasa	08:30:00
P006	Teknik Pengukuran	P003	Selasa	08:30:00
P001	Listrik Magnet	P004	Rabu	07:30:00
P002	Fisika Dasar	P004	Rabu	07:30:00
P003	Basis Data	P004	Rabu	07:30:00
P004	Mikrokontroler	P004	Rabu	07:30:00
P005	Matematika	P004	Rabu	07:30:00
P006	Teknik Pengukuran	P004	Rabu	07:30:00
P001	Listrik Magnet	P005	Kamis	09:00:00
P002	Fisika Dasar	P005	Kamis	09:00:00
P003	Basis Data	P005	Kamis	09:00:00
P004	Mikrokontroler	P005	Kamis	09:00:00
P005	Matematika	P005	Kamis	09:00:00
P006	Teknik Pengukuran	P005	Kamis	09:00:00
P001	Listrik Magnet	P006	Jun'at	09:00:00
P002	Fisika Dasar	P006	Jun'at	09:00:00
P003	Basis Data	P006	Jun'at	09:00:00
P004	Mikrokontroler	P006	Jun'at	09:00:00
P005	Matematika	P006	Jun'at	09:00:00
P006	Teknik Pengukuran	P006	Jun'at	09:00:00

36 rows in set (0.03 sec)

gambar 3.73 Hasil keluaran dari cross-join.

Apabila menggunakan query di atas, Anda akan mendapat tampilan sebanyak 36 baris. Jumlah tersebut berasal dari hasil perkalian antara recordset pada tabel matakuliah (pertama) dengan jumlah recordset pada tabel jadwal (kedua). Bentuk keluaran dua tabel tersebut memang sangat rumit. Untuk menyelesaikan permasalahan tersebut, Anda dapat menggunakan fungsi where. Berikut sintaksnya:

```
SELECT * FROM matakuliah
CROSS JOIN jadwal
WHERE matakuliah.kode_matkul=jadwal.kode_matkul;
```

Command Prompt - mysql -u latif -p

```
mysql> SELECT * FROM matakuliah CROSS JOIN jadwal
-> WHERE matakuliah.kode_matkul=jadwal.kode_matkul;
```

kode_matkul	matakuliah	kode_matkul	hari	jam
P001	Listrik Magnet	P001	Senin	07:30:00
P002	Fisika Dasar	P002	Senin	09:00:00
P003	Basis Data	P003	Selasa	08:30:00
P004	Mikrokontroler	P004	Rabu	07:30:00
P005	Matematika	P005	Kamis	09:00:00
P006	Teknik Pengukuran	P006	Jun'at	09:00:00

6 rows in set (0.31 sec)

Gambar 3.74 Hasil keluaran cross-join dengan fungsi where.

Setelah Anda jalankan perintah tadi, data yang ditampilkan berupa data tabel. Artinya, bukan lagi berupa perkalian dari dua tabel.

STRAIGHT JOIN

Straight join adalah salah satu operator untuk menjoinkan tabel. Fungsi straight join hampir sama dengan fungsi cross-join, hanya saja pada straight join tidak bisa diberikan klausa where.

```
SELECT * FROM matakuliah
```

```
STRAIGHT JOIN jadwal
```

Untuk contoh STRAIGHT JOIN, Anda gunakan lagi tabel matakuliah dan tabel jadwal.

Command Prompt - mysql -u latif -p

```
mysql> SELECT * FROM matakuliah STRAIGHT JOIN jadwal;
```

kode_matkul	matakuliah	kode_matkul	hari	jam
P001	Listrik Magnet	P001	Senin	07:30:00
P002	Fisika Dasar	P001	Senin	07:30:00
P003	Basis Data	P001	Senin	07:30:00
P004	Mikrokontroler	P001	Senin	07:30:00
P005	Matematika	P001	Senin	07:30:00
P006	Teknik Pengukuran	P001	Senin	07:30:00
P001	Listrik Magnet	P002	Senin	09:00:00
P002	Fisika Dasar	P002	Senin	09:00:00
P003	Basis Data	P002	Senin	09:00:00
P004	Mikrokontroler	P002	Senin	09:00:00
P005	Matematika	P002	Senin	09:00:00
P006	Teknik Pengukuran	P002	Senin	09:00:00
P001	Listrik Magnet	P003	Selasa	08:30:00
P002	Fisika Dasar	P003	Selasa	08:30:00
P003	Basis Data	P003	Selasa	08:30:00
P004	Mikrokontroler	P003	Selasa	08:30:00
P005	Matematika	P003	Selasa	08:30:00
P006	Teknik Pengukuran	P003	Selasa	08:30:00
P001	Listrik Magnet	P004	Rabu	07:30:00
P002	Fisika Dasar	P004	Rabu	07:30:00
P003	Basis Data	P004	Rabu	07:30:00
P004	Mikrokontroler	P004	Rabu	07:30:00
P005	Matematika	P004	Rabu	07:30:00
P006	Teknik Pengukuran	P004	Rabu	07:30:00
P001	Listrik Magnet	P005	Kamis	09:00:00
P002	Fisika Dasar	P005	Kamis	09:00:00
P003	Basis Data	P005	Kamis	09:00:00
P004	Mikrokontroler	P005	Kamis	09:00:00
P005	Matematika	P005	Kamis	09:00:00
P006	Teknik Pengukuran	P005	Kamis	09:00:00
P001	Listrik Magnet	P006	Jun'at	09:00:00
P002	Fisika Dasar	P006	Jun'at	09:00:00
P003	Basis Data	P006	Jun'at	09:00:00
P004	Mikrokontroler	P006	Jun'at	09:00:00
P005	Matematika	P006	Jun'at	09:00:00
P006	Teknik Pengukuran	P006	Jun'at	09:00:00

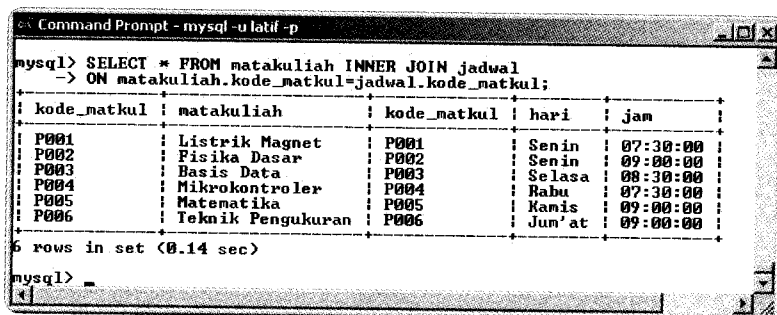
36 rows in set (0.67 sec)

Gambar 3.75 Hasil keluaran straight-join.

INNER JOIN

Inner join prinsipnya mirip dengan cross join yang memakai fungsi WHERE, hanya pada inner join fungsi WHERE digantikan ON. Kata kunci inner join terletak setelah klausa FROM dan sebelum filter jika memang ada filter. Ketika menggunakan join, hubungan antar tabel didefinisikan setelah kata kunci ON, bukan WHERE. Karena itulah syntax untuk join biasanya lebih panjang, tetapi juga lebih mudah terbaca karena kata WHERE hanya digunakan untuk memfilter hasil pengambilan datanya saja. Berikut ini contoh sintaksnya:

```
SELECT * FROM matakuliah  
  
INNER JOIN jadwal  
  
ON matakuliah.kode_matkul=jadwal.kode_matkul;
```



kode_matkul	matakuliah	kode_matkul	hari	jam
P001	Listrik Magnet	P001	Senin	07:30:00
P002	Fisika Dasar	P002	Senin	09:00:00
P003	Basis Data	P003	Selasa	08:30:00
P004	Mikrokontroler	P004	Rabu	07:30:00
P005	Matematika	P005	Kamis	09:00:00
P006	Teknik Pengukuran	P006	Jum'at	09:00:00

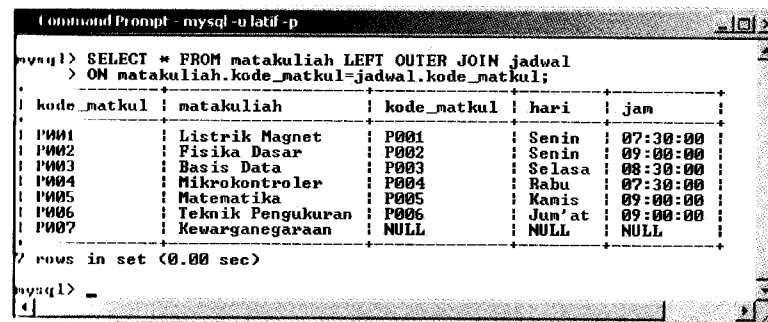
Gambar 3.76 Hasil keluaran dari inner join menggunakan fungsi on.

LEFT [OUTER] JOIN

Seandainya ingin menampilkan dua tabel, tabel pertama bisa dianggap sebagai tabel kiri (left) dan tabel kedua sebagai tabel kanan (right). Dengan menggunakan join left outer ini, operator left join menampilkan seluruh data pada tabel kiri. Apabila salah satu kolom anggota pada kolom kanan tidak memiliki hubungan dengan tabel kiri, data pada tabel kanan akan diisi NULL. Nilai NULL tersebut berarti tidak ada relasi antar tabel kanan dan kiri.

Dituk lebih jelas, berikut ini contoh sintaksnya:

```
SELECT * FROM matakuliah  
  
LEFT OUTER JOIN jadwal  
  
ON matakuliah.kode_matkul=jadwal.kode_matkul;
```



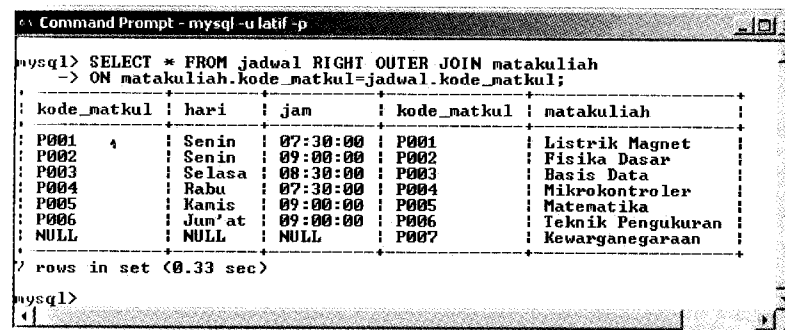
kode_matkul	matakuliah	kode_matkul	hari	jam
P001	Listrik Magnet	P001	Senin	07:30:00
P002	Fisika Dasar	P002	Senin	09:00:00
P003	Basis Data	P003	Selasa	08:30:00
P004	Mikrokontroler	P004	Rabu	07:30:00
P005	Matematika	P005	Kamis	09:00:00
P006	Teknik Pengukuran	P006	Jum'at	09:00:00
P007	Kewarganegaraan	NULL	NULL	NULL

Gambar 3.77 Hasil keluaran dari left outer join.

RIGHT [OUTER] JOIN

Right [outer] join merupakan kebalikan left [outer] join. Pada right [outer] join data yang berada pada tabel kedua (RIGHT) akan ditampilkan semua. Apabila pada tabel pertama (LEFT) tidak ada relasi pada tabel kedua, akan diisi data kosong atau NULL. Berikut ini contoh sintaksnya:

```
SELECT * FROM jadwal  
  
RIGHT OUTER JOIN matakuliah  
  
ON matakuliah.kode_matkul=jadwal.kode_matkul;
```



kode_matkul	hari	jam	kode_matkul	matakuliah
P001	Senin	07:30:00	P001	Listrik Magnet
P002	Senin	09:00:00	P002	Fisika Dasar
P003	Selasa	08:30:00	P003	Basis Data
P004	Rabu	07:30:00	P004	Mikrokontroler
P005	Kamis	09:00:00	P005	Matematika
P006	Jum'at	09:00:00	P006	Teknik Pengukuran
NULL	NULL	NULL	P007	Kewarganegaraan

Gambar 3.78 Hasil keluaran dari right outer join.

IDENTIFIER

MySQL memungkinkan pemberian nama yang terdiri dari satu atau lebih identifier. Komponen dari nama yang multibagian harus dipisahkan oleh karakter titik. Bagian pertama berperan sebagai qualifier yang mempengaruhi konteks di mana identifier terakhir diinterpretasikan.

Di MySQL, Anda dapat mengacu kolom tabel menggunakan beberapa bentuk berikut:

Tabel 3.1 Bentuk kolom tabel

Referensi Kolom	Arti
Col_name	Kolom col_name di mana tabel yang digunakan di statement mengandung nama kolom tersebut
Tbl_name.col_name	Kolom col_name dari tabel bernama tbl_name dari database default
Db_name.tbl_name.col_name	Kolom col_name dari tabel tbl_name dari database db_name

Jika ada komponen di nama-nama yang multibagian memerlukan quote, quote dilakukan secara terpisah dan bukan keseluruhan. Jadi, penulisan yang benar adalah `my-table`.`my-column`, dan bukan `my-table.my-column`.

Reserved word yang mengikuti tanda titik adalah identifier sehingga tidak perlu dikasih quote.

PENGEMBANGAN MySQL

Bab ini akan membahas:

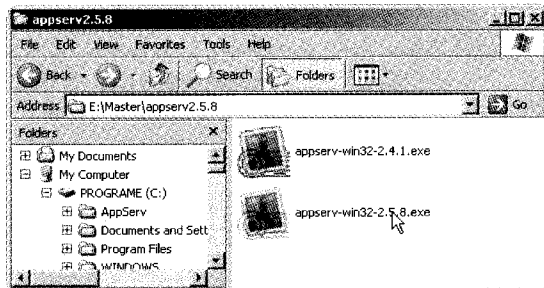
- Cara mengoneksikan MySQL.
- Bagaimana menampilkan database.
- Memahami pembuatan aplikasi.

Pada bab ini akan dibahas tentang cara mengoneksikan database pada MySQL dengan program-program penampil, di antaranya adalah PHP dan Visual Basic. Di sini juga akan dibahas cara menampilkan database dari MySQL sampai pembuatan aplikasi dengan menggunakan database MySQL.

INSTALASI PHP DAN APACHE

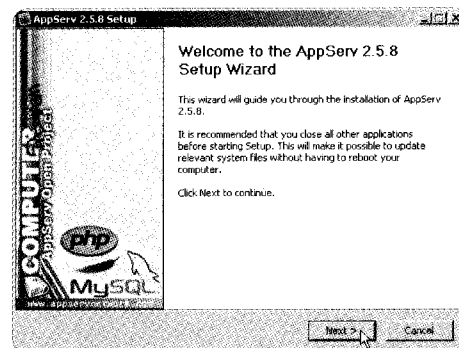
Sebelum Anda melakukan menerapkan database MySQL, Anda harus menginstall terlebih dahulu program PHP dan apache sebagai web server. Untuk mempersingkat kata, langsung saja lakukan proses instalasi PHP dan apache. Untuk memudahkan Anda, di sini menggunakan appserv yang memiliki satu bundel PHP dan apache. Anda bisa mendownload software-nya di <http://www.appservnetwork.com>. Langkah-langkah instalasinya adalah sebagai berikut:

1. Buka folder yang berisi instalasi appserv. Klik dua kali appserv 2.5.8.



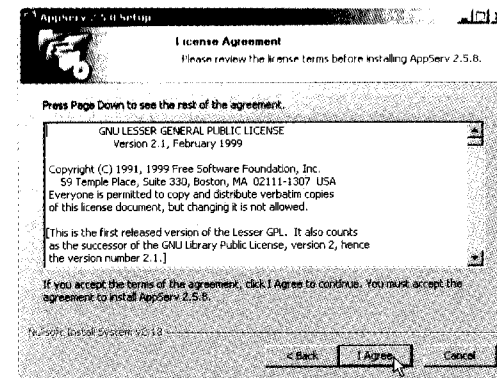
Gambar 4.1 Direktori di mana file appserv disimpan.

2. Klik **Next**.



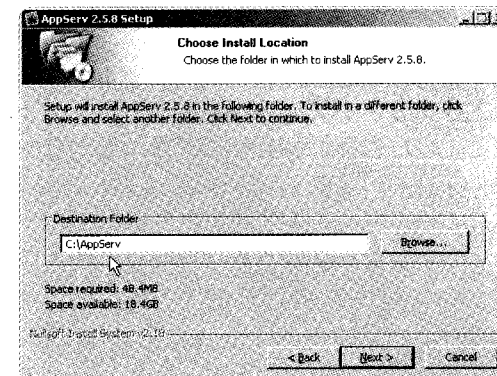
Gambar 4.2 Tampilan pertama setup wizard.

3. Klik **I Agree**.



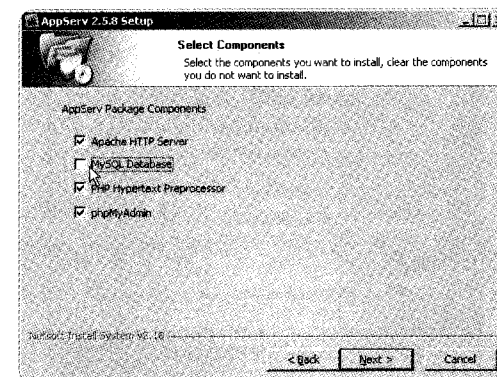
Gambar 4.3 Persetujuan dengan pihak pembuat

4. Pilih di mana Anda akan meletakkan file instalasi lalu klik **Next**.



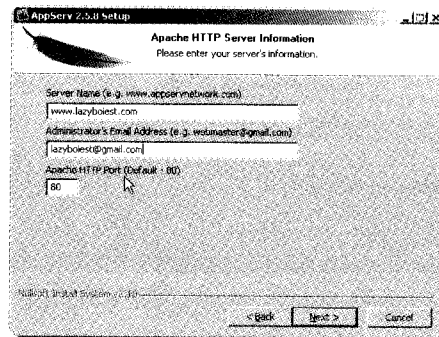
Gambar 4.4 File setelah instalasi diletakkan.

5. Hilangkan ceklis pada **MySQL Database** karena Anda sudah menginstal MySQL sebelumnya. Kemudian klik **Next**



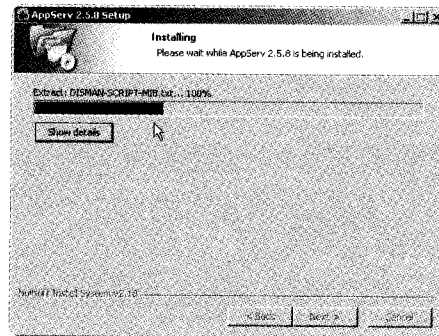
Gambar 4.5 Memilih program apa yang akan diinstal.

- Isi nama server yang Anda sukai, kemudian alamat email Anda sebagai administrator dan apache HTTP port biarkan secara default yaitu 80. Kemudian klik **Next**.



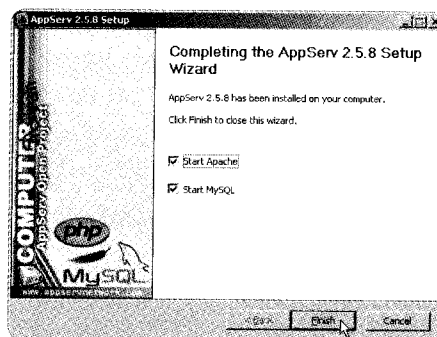
Gambar 4.6 Setting web server.

- Saat proses instalasi berlangsung.



Gambar 4.7 Proses instalasi.

- Instalasi selesai, cek pada **Start Apache** dan **Start MySQL** biarkan saja agar dapat berjalan saat instalasi komplit. Klik **Finish** untuk mengakhiri instalasi.



Gambar 4.8 Setup wizard komplet.

INTERAKSI DATABASE MYSQL DENGAN PHP

Pada bab sebelumnya sudah dibahas bahwa MySQL merupakan database yang berbasis server. Anda dapat menggunakan database MySQL apabila Anda memiliki hak izin akses di dalamnya. Hal ini seperti halnya pada saat Anda hendak menggunakan klien mysql untuk masuk pada server MySQL.

Dengan penggunaan program PHP, Anda dapat mengakses database MySQL dengan terlebih dahulu melakukan koneksi atau login ke server. Fungsi yang digunakan untuk login ke server atau melakukan koneksi ke dalam server adalah `mysql_connect()`. Untuk lebih jelasnya, lihat program berikut:

```
<html>

<head>

<title> Koneksi Database

</title>

</head>

<body>

<?

    $hostname="localhost";

    $username="root";

    $password="";

    if (!$conn=mysql_connect($hostname,$username,$password))

    {

        echo mysql_error();return 0;

    }

    else{
```

```
echo"koneksi ke database telah berhasil";
```

```
}
```

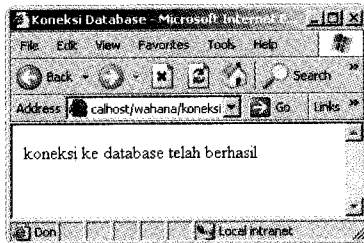
```
??
```

```
</body>
```

```
</html>
```

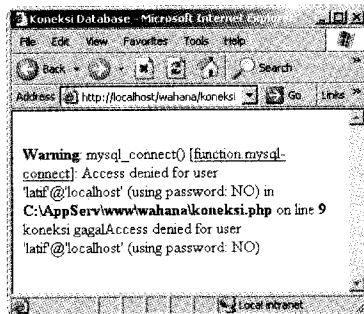
Setelah Anda mengetikkan script di atas, simpanlah dengan nama **koneksi.php** dan simpan pada direktori **C:\appserv\www**. Pada program di atas, Anda melakukan koneksi server MySQL dengan menggunakan username "latif" dan mempunyai password "insel".

Karena server MySQL yang diakses berada pada komputer lokal (satu sesi dengan web server yang sedang digunakan), Anda dapat menggunakan "localhost" sebagai nama hostname-nya. Untuk dapat menampilkan pada jendela browser, ketikkan <http://localhost/koneksi.php> pada address bar. Apabila koneksi yang Anda lakukan berhasil, browser akan tampak seperti pada gambar di bawah ini.



Gambar 4.9 Tampilan koneksi ke server berhasil

Apabila Anda gagal dalam menghubungkan ke server, tampilan yang Anda dapatkan kurang lebih tampak seperti di bawah ini.



Gambar 4.10 Tampilan koneksi ke server gagal.

MENGAKTIFKAN DATABASE

Setelah login dinyatakan berhasil, dan Anda telah berada pada prompt MySQL. Anda dapat memilih database mana yang akan digunakan untuk bekerja. Pada PHP Anda dapat menggunakan fungsi `mysql_select_db()` sebagai operasi aktivasi database. Fungsi tersebut sama dengan perintah `USE` jika Anda bekerja menggunakan prompt MySQL. Untuk lebih jelasnya, lihat program berikut:

```
<html>

<head>

<title> Mengaktifkan database</title>

</head>

<body>

<?

    $hostname="localhost";

    $username="root";

    $password="";

    $db="mahasiswa";

    if (!$conn=mysql_connect($hostname,$username,
    $password))

    {

        echo mysql_error();return 0;

    }

    if (!$mysql_select_db($db))

    {

        echo mysql_error();return 0;

    }

    else{

        echo"database telah berhasil dibuka";
```

```

?>
</body>
</html>

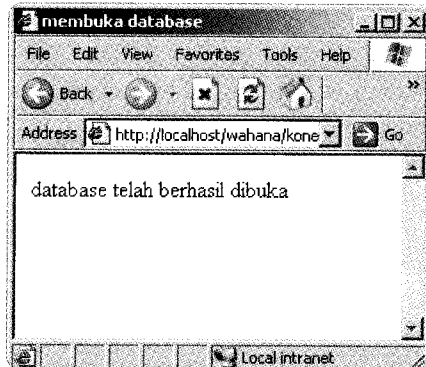
```

Keterangan program di atas adalah sebagai berikut:

Pada program di atas terdapat perintah `mysql_select_db(wahana, $koneksi)`, perintah ini digunakan untuk mengaktifkan database.

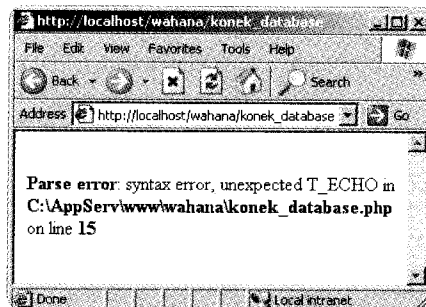
Selain perintah tersebut, juga terdapat perintah `or die ("database gagal dibuka".mysql_error());` yang digunakan untuk menampilkan pesan apabila terjadi kesalahan.

Jika Anda berhasil membuka database, Anda akan mendapatkan tampilan seperti di bawah ini.



Gambar 4.11 Tampilan database telah berhasil diaktifkan.

Apabila ada kesalahan dalam mengetikkan program, Anda akan mendapatkan tampilan kurang lebih seperti gambar di bawah ini.



Gambar 4.12 Tampilan kegagalan dalam mengaktifkan database.

MENAMPILKAN ISI DARI DATABASE

Anda telah berhasil mengoneksikan database server dan berhasil mengaktifkan database, sekarang Anda harus menampilkan isi dari database tersebut. Untuk menampilkan isi dari database, Anda memerlukan SQL query yang dapat Anda jalankan melalui program PHP. Untuk menjalankan SQL query pada PHP, Anda dapat menggunakan fungsi `mysql_query()`. Untuk lebih jelasnya, perhatikan contoh program berikut:

```

<html>

<head>

<title>menampilkan isi database</title>

</head>

<body>

<?php

$koneksi=mysql_connect("localhost","root","");

$db=mysql_select_db("mahasiswa",$koneksi);

$sql="select * from pelajaran";

$qry=mysql_query($sql, $koneksi) or die
("terjadi kesalahan");

?>

</body>

</html>

```

Dengan perintah query SQL seperti program di atas, tidak akan menghasilkan bentuk keluaran. Supaya datanya dapat ditampilkan pada jendela browser, Anda harus memisahkan datanya ke dalam bentuk array, kemudian menampilkannya dengan perintah `echo`. Untuk menampilkan data dari datanya jalankan program berikut:

```

<html>

<head>

<title>menampilkan isi database</title>

</head>

<body>

<?php

$koneksi=mysql_connect("localhost","root","");

$db=mysql_select_db("mahasiswa",$koneksi);

$sql="select * from mahasiswa";

$qry=mysql_query($sql, $koneksi) or die
("terjadi kesalahan");

$no=0;

while($baris=mysql_fetch_array($qry,MYSQL_NUM))
{

$no+=1;

echo"<table>

        <tr>

                <td>No</td><td>$no</td>

        </tr>

        <tr>

                <td>Kode Pelajaran</
td><td>$baris[0]</td>

        </tr>

        <tr>

```

```

        <td>Nama Pelajaran /
        <td>$baris[1]</td>

        </tr>

</table>";

}

}

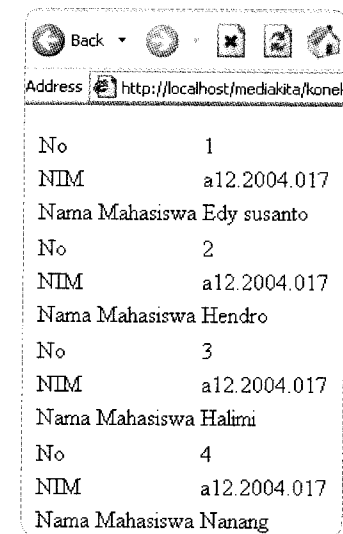
</body>

</html>

```

terlihat pada program di atas, untuk memecahkan data hasil query, PHP memiliki fungsi `mysql_fetch_array`. Dengan menggunakan fungsi tersebut, hasil perintah `SELECT` akan dijadikan tumpukan (array). Dari program di atas juga terdapat operator perulangan `while`. Dengan operator tersebut, Anda dapat mengeluarkan seluruh baris data yang dihasilkan dari perintah `SELECT` di atas. Hasil dari `SELECT` akan Anda masukkan ke dalam variable `$baris` sebagai data array.

Dengan menjalankan program di atas, Anda akan mendapatkan tampilan pada browser seperti di bawah ini.



No	1	
NIM	a12.2004.017	
Nama Mahasiswa	Edy susanto	
No	2	
NIM	a12.2004.017	
Nama Mahasiswa	Hendro	
No	3	
NIM	a12.2004.017	
Nama Mahasiswa	Halimi	
No	4	
NIM	a12.2004.017	
Nama Mahasiswa	Nanang	

Gambar 4.13 Tampilan dari database mahasiswa tabel mahasiswa.

Membuat program masukan pada database

Selain menggunakan prompt MySQL untuk memasukkan data, Anda juga dapat memasukkan data dengan program PHP. Jika Anda memasukkan data melalui prompt MySQL, Anda akan kesulitan dan memakan waktu yang lama apabila data yang dimasukkan sangat banyak. Solusi paling cepat adalah dengan membuat form. Salah satunya adalah dengan menggunakan PHP. Untuk lebih jelasnya, langsung saja perhatikan program berikut untuk membuat form:

```
<html>

<head>

<title>Insert_Database</title>

</head>

<body>

<?php

$koneksi=mysql_connect("localhost","root","");

$db=mysql_select_db("mahasiswa",$koneksi);

?>

<form action="" method="get">

<table width="422" border="1">

<tr>

<th width="412" scope="col">MENAMBAH
RECORD</th>

</tr>

</table>

<table width="313" border="0">

<tr>
```

```
<th width="123" scope="col">&nbsp;</th>

<th width="178" scope="col">&nbsp;</th>

</tr>

<tr>

<td width="124">ID Mahasiswa:</td>

<td width="250"><input name="id_mahasiswa"
type="text" size="10" maxlength="10"></td>

</tr>

<tr>

<td>Nama</td>

<td><input name="nama" type="text"
size="20" maxlength="20"></td>

</tr>

<tr>

<td valign="top">Alamat</td>

<td><textarea name="alamat" cols="30"
rows="3" id="alamat"></textarea></td>

</tr>

<tr>

<td>Jenis Kelamin:</td>

<td><select name="jenis_kelamin"
style="font-size:12px;">

<option
value="L" selected>L</option>

<option
```

```
value="L" type="option"
```

```
</select>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td><div align="center">
```

```
<input name="simpan" type="submit"
id="simpan" value="Simpan"></div>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</form>
```

```
<?php
```

```
if ($simpan)
```

```
{
```

```
mysql_query("insert into mahasiswa values('$sid_
mahasiswa','$nama','$alamat','$jenis_
kelamin')");
```

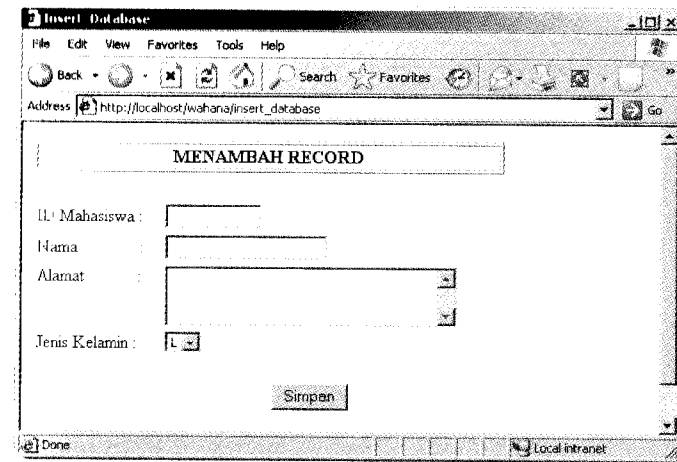
```
}
```

```
?>
```

```
</body>
```

```
</html>
```

Setelah membuat program di atas, simpan dengan nama file **insert_database.php** kemudian jalankan dengan browser. Anda akan mendapatkan tampilan di bawah ini:



Gambar 4.14 Tampilan menambah record

Setelah menjalankannya di browser, isi semua kolom yang akan dimasukkan ke tabel mahasiswa. Sebagai contoh, Anda ingin menambah record ID Mahasiswa dengan 1001, Nama Abdul Latif, Alamat Jl. Tirto Hurodo barat No.31, Jenis Kelamin Anda tinggal memilih L/P. Setelah klik simpan, record pada tabel mahasiswa akan bertambah.

MEMBUAT PROGRAM HAPUS PADA DATABASE

Selain untuk menambah record, program PHP juga dapat menghapus record. Jadi, Anda tidak perlu report lagi apabila Anda salah dalam memasukkan atau menambah record. Anda dapat langsung menghapusnya dengan program PHP. Untuk dapat menghapus record, buatlah program seperti berikut:

```
<html>
```

```
<head>
```

```
<title>Hapus Record</title>
```

```
</head>
```

```

· body ·

<?php

$koneksi=mysql_connect("localhost","root","");

$db=mysql_select_db("mahasiswa",$koneksi);

?>

<form action="" method="get">

    <table width="422" border="1">

        <tr>

            <th width="412" scope="col">MENGHAPUS
RECORD </th>

        </tr>

    </table>

    <table width="313" border="0">

        <tr>

            <th width="123" scope="col">&nbsp;</th>

            <th width="178" scope="col">&nbsp;</th>

        </tr>

    </table>

    <table width="384" border="0">

        <tr>

            <td width="124">ID Mahasiswa : </td>

            <td width="250"><label>

```

```

        <input name="id mahasiswa" type="text"
size="10" maxlength="10">

        </label>    </td>

    </tr>

    <tr>

        <td>Nama

        </td>

        <td><label>

            <input name="nama" type="text"
size="20" maxlength="20">

        </label>    </td>

    </tr>

    <tr>

        <td valign="top">&nbsp;</td>

        <td><label></label>    </td>

    </tr>

    <tr>

        <td>&nbsp;</td>

        <td>&nbsp;</td>

    </tr>

    <tr>

        <td>&nbsp;</td>

        <td>&nbsp;</td>

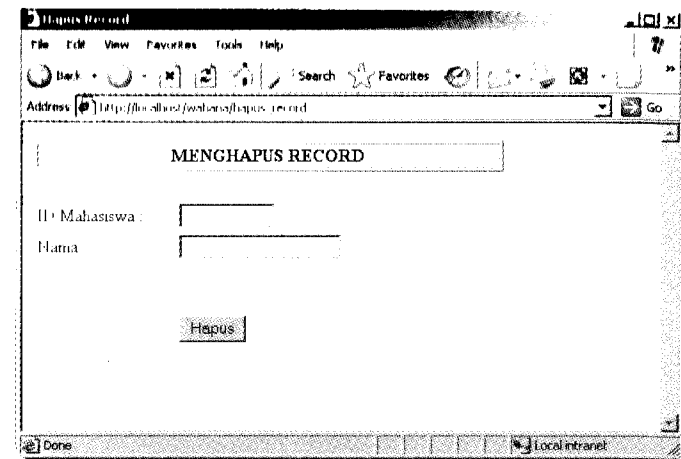
    </tr>

```

```
  || |  | |

```

Setelah membuat program di atas, simpan dengan nama **hapus_record.php**. Kemudian jalankan pada browser. Anda akan mendapatkan tampilan seperti berikut ini.



Gambar 4.15 Tampilan hapus record.

Setelah menjalankannya di browser, isi ID Mahasiswa dan Nama mahasiswa yang akan dihapus dari record. Sebagai contoh, Anda ingin menghapus ID Mahasiswa 1001 dan yang mempunyai Nama Abdul Latif, kemudian tekan hapus. Dengan demikian, id mahasiswa 1001 dengan nama Abdul Latif akan terhapus dari tabel mahasiswa.

Membuat program edit pada database

Selain untuk menambah dan menghapus record, program PHP juga dapat mengedit atau meng-update isi dari tabel. Silakan coba kode berikut:

```

<html>
<head>
<title>Ubah_Record</title>
</head>
<body>
<?php
$koneksi=mysql_connect("localhost","root","");
$db=mysql_select_db("mahasiswa",$koneksi);
?>

```

```

form action "" method "get"

<table width="422" border="1">

  <tr>

    <th width="412" scope="col">MEPERBARUI
RECORD </th>

  </tr>

</table>

<table width="313" border="0">

  <tr>

    <th width="123" scope="col">&nbsp;</th>

    <th width="178" scope="col">&nbsp;</th>

  </tr>

</table>

<table width="384" border="0">

  <tr>

    <td width="124">ID Mahasiswa : </td>

    <td width="250"><label>

      <input name="id_mahasiswa" type="text"
size="10" maxlength="10">

    </label>    </td>

  </tr>

  <tr>

    <td>Nama

```

```

  </td>

  <td><input name="nama" type="text"
size="20" maxlength="20"> </td>

</tr>

<tr>

  <td valign="top">Alamat

  </td>

  <td><label>

    <textarea name="alamat" cols="30"
rows="3" id="alamat">

    </textarea>

  </label>    </td>

</tr>

<tr>td>Jenis Kelamin : </tr> <td>

  <select name="jenis_kelamin"
style="font-size:12px;">

    <option
value="L" selected>L</option>

    <option
value="P">P</option>

  </select>

  </td>

</tr>

```

```

        <td>&nbsp;</td>

        <td>&nbsp;</td>

    </tr>

    <tr>

        <td>&nbsp;</td>

        <td><div align="center"><label>

            <input name="ubah" type="submit"
            id="ubah" value="Update">

            </label></div>

        </td>

    </tr>

</table>

<p>&nbsp;</p>

</form>

<?php
if ($ubah)
{

    mysql_query("UPDATE mahasiswa SET

        id_mahasiswa=' $id_mahasiswa',nama=' $na
ma',alamat=' $alamat',jenis_kelamin=' $jenis_
kelamin'");
}

?>

```

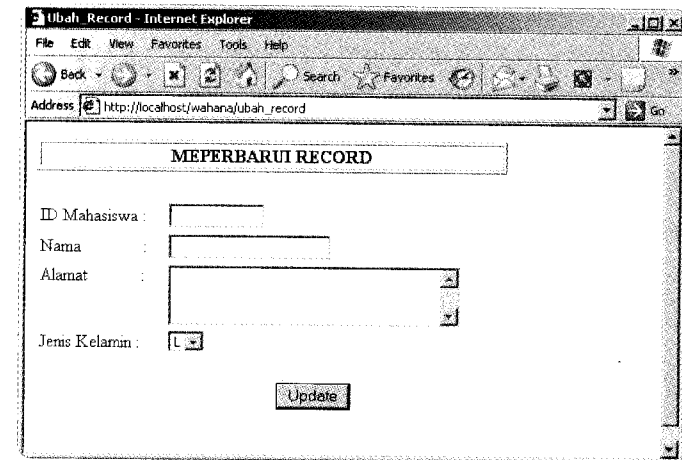
```

    </body>

</html>

```

Setelah membuat program di atas, simpanlah dengan nama `ubah_record.php`. Setelah itu, jalankan dengan browser. Anda akan mendapatkan tampilan seperti berikut.



gambar 4.16 Tampilan memperbaiki record.

Setelah menjalankannya pada browser, isi data yang Anda ingin ubah. Sebagai contoh, Anda ingin mengubah alamat mahasiswa menjadi di jl. Kenteng sari no.1A, yang sudah Anda isikan pada tabel mahasiswa pada sub bab menambah record. Isi ID Mahasiswa dengan 1001, Nama isi dengan Abdul Latif, alamat isi dengan L. Kemudian klik tombol **Update**, alamatnya akan berubah.

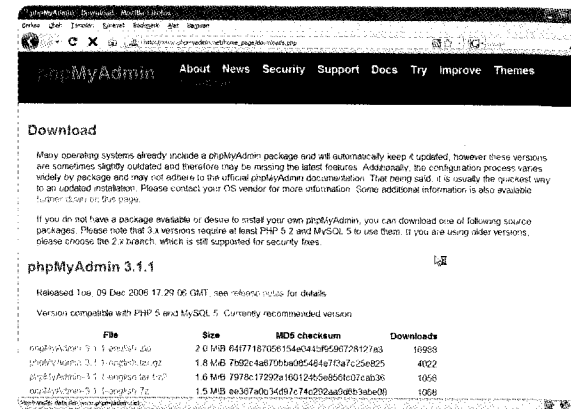
PENGUNAAN PHPMyAdmin UNTUK WEB DEVELOPMENT

Bab ini akan membahas:

- Instalasi PHPMyAdmin.
- Mengenal userinterface PHPMyAdmin.
- Membuat dan mengolah database dengan interface web.
- Operasi tabel dengan PHPMyAdmin.

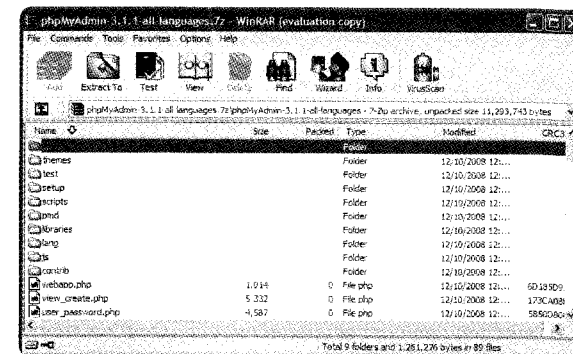
2. MySQL 5.0 atau lebih baru. Di phpMyAdmin 3.0.x, versi MySQL yang didukung adalah mulai versi MySQL 5.0.1. PHPMysqlAdmin juga bisa konek menggunakan ekstensi klasik dari PHP dan juga ekstensi

- 1 Web browser dengan cookie diaktifkan.



Gambar 5.1 Halaman download PHPMyAdmin.

File file installer hanya berupa file .php yang langsung diletakkan di folder dan nantinya bisa diakses untuk mengakses MySQL server yang ada di server lokal tersebut. Tergantung konfigurasi server, tetapi beberapa server harus mengubah konfigurasi php.ini agar bisa berjalan dengan optimal.



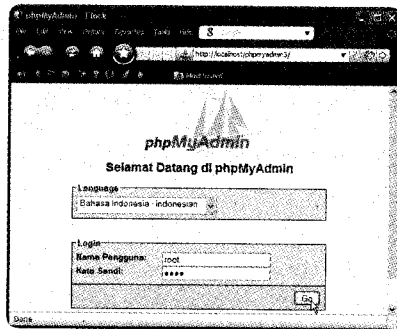
Gambar 5.2 File PHPMyAdmin All language versi 3 dibuka di WinRAR.

Jika PHPMyAdmin diinstal oleh software Appserv atau XAMPP, untuk mengakses PHPMyAdmin tersebut dapat \$tampil=mysql_query("SELECT * FROM mahasiswa", \$koneksi)

```
or die("query gagal".mysql_error());
```

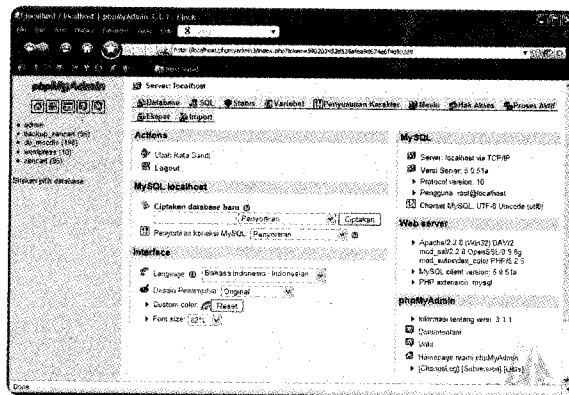
menggunakan URL <http://localhost/PHPMyAdmin> atau tergantung kepada setting server yang digunakan.

Pertama, pilih bahasa di **Language**. Untuk Indonesia sudah ada, tinggal memilih **Bahasa Indonesia-Indonesian** di kotak **Language**. Login dengan username dan password untuk MySQL server tersebut. Username diisi di **Name pengguna**, sementara password di **Kata Sandi**. Klik **Go** untuk masuk ke PHPMyAdmin.



Gambar 5.3 Halaman login untuk masuk ke PHPMyAdmin versi 3.

Halaman awal setelah login di phpMyAdmin versi 3 seperti berikut. Ada 2 frame, frame kiri menampilkan database yang ada (jika sudah ada), sementara di bagian kanan terdapat beberapa kotak untuk pengaturan atau manajemen MySQL server.



Gambar 5.4 MySQL server.

DASBOR PHPMYADMIN

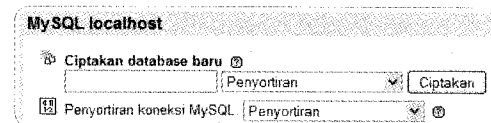
Dasbor PHPMyAdmin terdapat beberapa kotak untuk melakukan fungsi manajemen PHPMyAdmin secara interaktif menggunakan interface web. Tiap kotak di dasbor ada fungsinya sendiri-sendiri.

Yang pertama adalah kotak Actions. Di sini ada 2 bagian, yaitu Ubah Kata Sandi untuk mengganti password dan Logout untuk keluar dari PHPMyAdmin ini.



Gambar 5.5 Kotak Actions.

Adapun yang kedua adalah MySQL localhost. Kotak ini menangani database MySQL dan digunakan untuk membuat database baru.



Gambar 5.6 Kotak MySQL localhost.

Kotak ketiga adalah interface yang berfungsi mengubah tampilan interface MySQL. Dari mulai bahasa/language, desain tampilan/theme, warna custom, dan ukuran font bisa diubah di kotak ini.

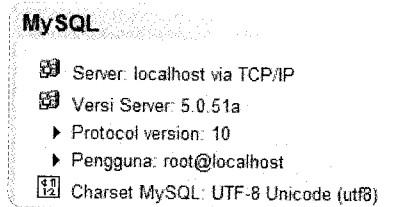


Gambar 5.7 Kotak interface di dasbor PHPMyAdmin.

Kotak keempat adalah MySQL. Di kotak MySQL ini terdapat informasi mengenai kondisi server MySQL, versi MySQL server, protokol, user yang sedang login, dan set karakter (charset) yang digunakan. Kotak ini tak bisa diedit nilainya atau hanya bisa dilihat.

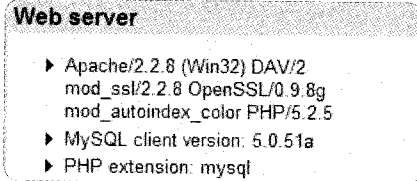
User yang dimaksud di PHPMyAdmin ini adalah user dari MySQL server, bukan user dari PHPMyAdmin. PHPMyAdmin tidak memiliki sistem tersendiri untuk menyimpan database-nya. Ketika user login ke PHPMyAdmin, username, dan password langsung digunakan untuk login

ke MySQL server. PHPMyAdmin tidak memiliki manajemen account terpisah sehingga user-user yang bisa login haruslah user yang sudah terdaftar di MySQL secara valid.



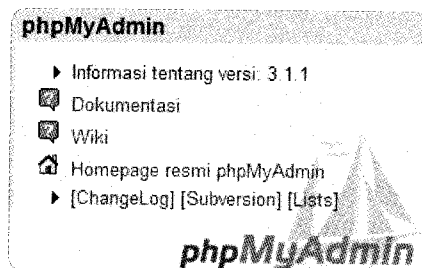
Gambar 5.8 Kotak MySQL di dasbor PHPMyAdmin.

Kotak kelima adalah web server yang menampilkan kondisi web server (Apache atau IIS) yang dipakai. Jika menggunakan server Apache, ditampilkan juga nama modul-modul yang berkaitan dengan MySQL serta ekstensi PHP.



Gambar 5.8 Kotak MySQL di dasbor PHPMyAdmin.

Kotak terakhir adalah PHPMyAdmin. Di sini dijelaskan versi PHPMyAdmin beserta link-link untuk menuju ke website resmi phpMyAdmin, wiki, serta dokumentasi.



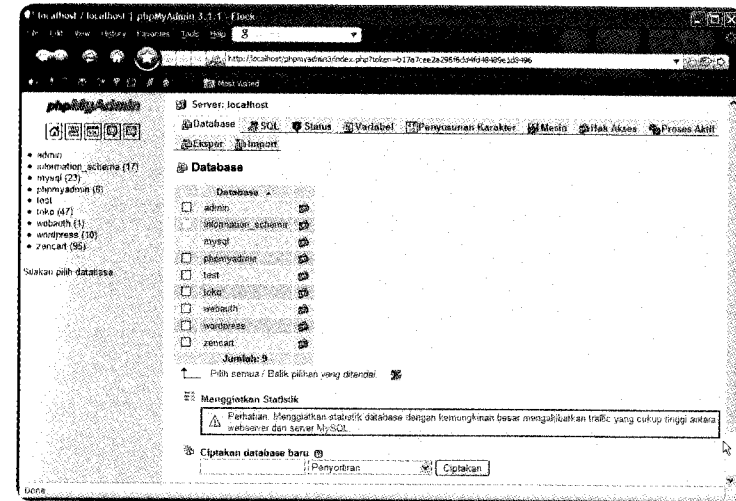
Gambar 5.10 Kotak PHPMyAdmin di dasbor PHPMyAdmin.

MENGENAL TAB PHPMYADMIN

Dari halaman dasbor dan halaman lainnya di PHPMyAdmin, terdapat tab-tab di bagian atasnya. Ini berguna untuk membuka halaman-

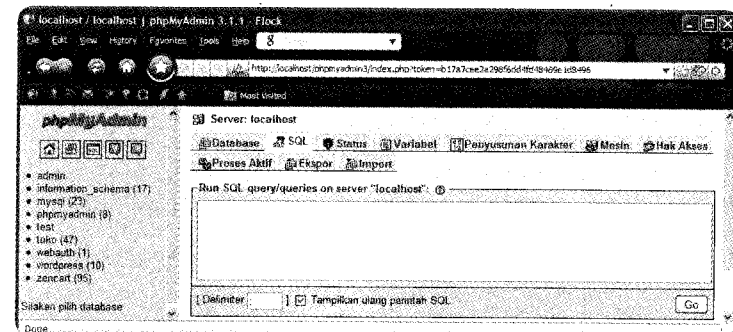
halaman yang memiliki fungsi khusus di PHPMyAdmin. Jika yang terbuka adalah server MySQL, ada 10 tab, mulai Database sampai Import.

Tab database menampilkan semua database yang ada di server MySQL yang bersangkutan. Satu MySQL server bisa memiliki lebih dari satu database, di samping sudah ada beberapa database bawaan, seperti information_schema, dan phpMyAdmin. Database-database yang ada ditampilkan dalam format tabel 3 kolom di mana kolom keduanya menampilkan nama database yang bersangkutan.



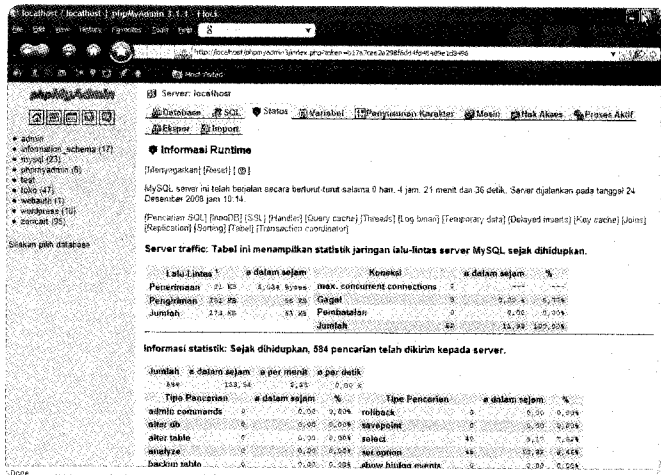
Gambar 5.11 Nama database ditampilkan di tab Database.

Tab kedua adalah tab SQL yang di dalamnya terdapat kotak Run SQL query/queries di mana kotak tersebut nantinya untuk menjalankan sql statement. Sql statement bisa digunakan untuk mengakses database ataupun tabel.



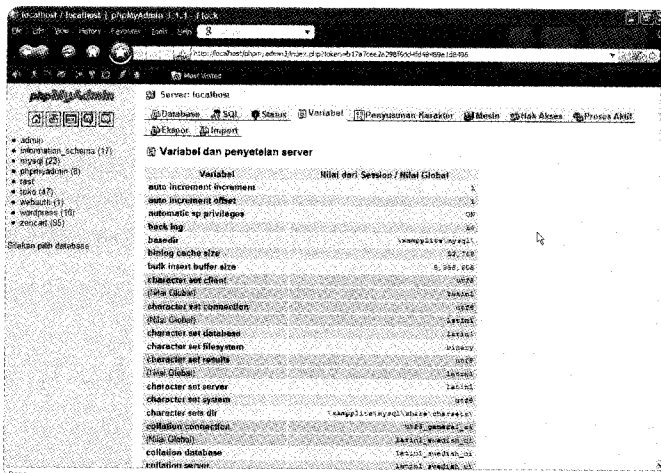
Gambar 5.12 Tab SQL yang menampilkan kotak Run SQL query/queries.

Tab ketiga adalah tab **Status**. Tab ini menampilkan informasi run time dan informasi status lainnya berkaitan dengan PHPMyAdmin ini. Misalnya, PHPMyAdmin sudah dijalankan berapa lama, jumlah lalu lintas data, dan informasi statistik sejak PHPMyAdmin dihidupkan. Ada banyak informasi yang ditampilkan di halaman ini, semuanya bertujuan untuk mengetahui status PHPMyAdmin.



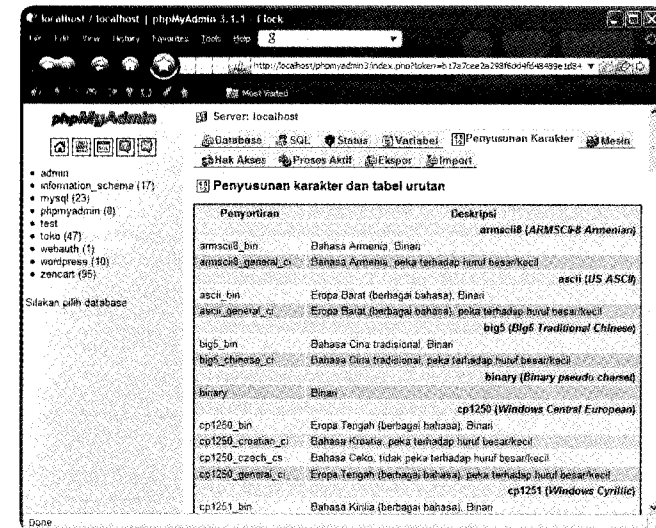
Gambar 5.13 Informasi di tab Status di PHPMYAdmin.

Tab keempat adalah tab Variabel yang mengatur berbagai variabel dan informasi-informasi penyetelan MySQL server. Tab ini hanya berisi informasi dan tidak bisa mengubah penyetelan di sini.



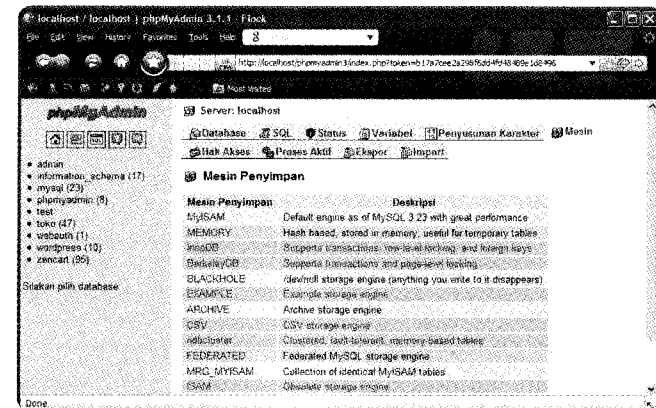
Gambar 5.14 Tab Variabel di PHPMyAdmin.

Tab **Penyusunan karakter** menunjukkan penjelasan mengenai karakter-karakter yang ada di PHPMyAdmin dan juga deskripsi dari berbagai jenis karakter yang ada. Tab ini juga berisi informasi saja dan tidak bisa melakukan penyetingan di sini.



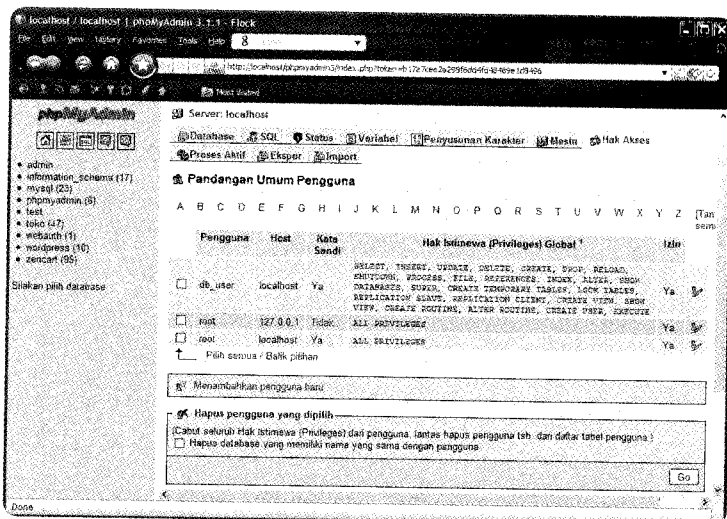
Gambar 5.15 Tab Penyusunan karakter di PHPMyAdmin.

Tabel 1.1 menjelaskan berbagai mesin penyimpanan yang didukung oleh MySQL server. Mesin penyimpanan yang terinstal di server terlihat berwarna biru, sementara yang tidak didukung oleh server berwarna hitam putih. Mesin penyimpanan (storage engine) yang ada merupakan bawaan instalasi MySQL server. Jika ingin menambah mesin penyimpanan yang lain, bisa diinstal secara terpisah di MySQL server.



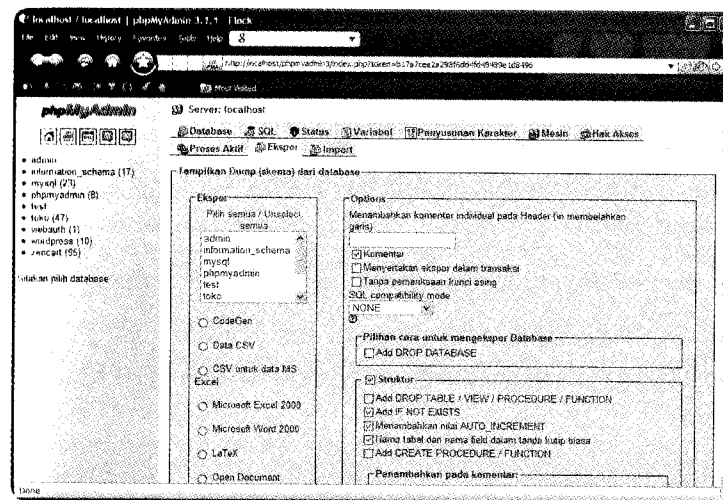
Gambar 5.16 Tab Mesin penyimpanan menampilkan berbagai mesin penyimpanan yang ada di MySQL server.

Tab **Hak Akses** menjelaskan user user yang sudah terdaftar di MySQL server. User bisa dihapus atau diedit atributnya di tab Hak Akses ini. User standar di MySQL server adalah root.



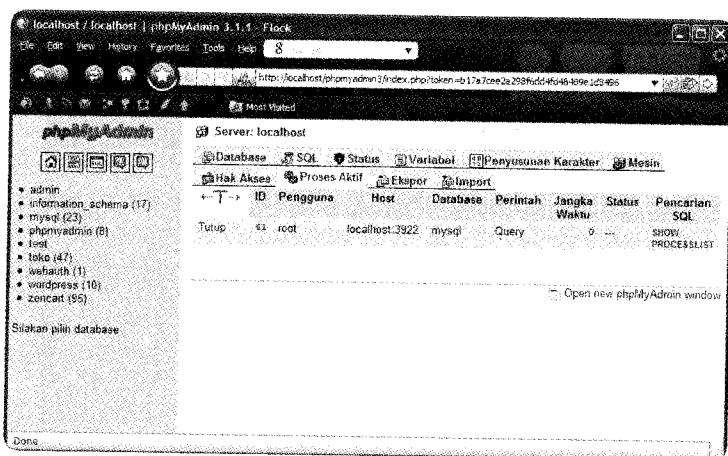
Gambar 5.17 Halaman MySQL server.

Tab **Ekspor** menampilkan halaman untuk diekspor. Ekspor adalah sebuah kegiatan memindahkan file-file database beserta isi di dalamnya ke dalam sebuah format file tertentu sehingga isi file tersebut nantinya bisa diimpor. Tujuannya adalah untuk menghindari adanya kehilangan file (backup).



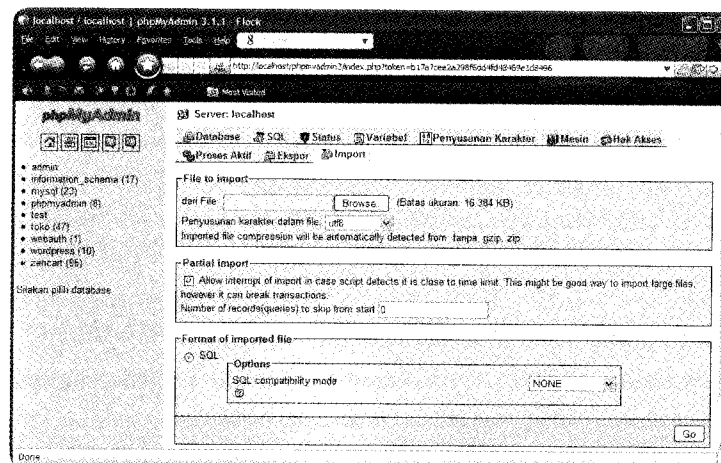
Gambar 5.19 File backup.

Tab **Proses aktif** menjelaskan proses-proses yang sedang dijalankan/ aktif di server MySQL. Misalnya, ada lebih dari satu user yang sedang mengakses atau ada aplikasi yang menggunakan database, akan ditampilkan di sini.



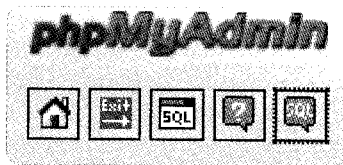
Gambar 5.18 Proses-proses yang sedang aktif di MySQL terlihat di tab Proses Aktif.

Tab **Import** menampilkan kotak-kotak untuk mengimpor isi database dari file tertentu sehingga bisa dimasukkan ke dalam file MySQL. Tujuannya agar file backup bisa dikembalikan ke MySQL server, baik itu di server aslinya atau pun untuk memindahkan data ke server lain.



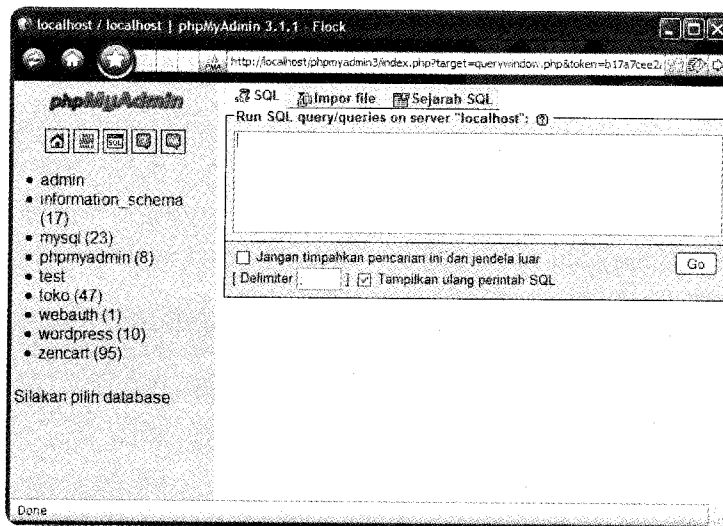
Gambar 5.20 Tab Import di PHPMyAdmin.

Di frame sebelah kiri, di bawah tulisan PHPMyAdmin, ada 5 tombol untuk mengakses menu-menu tertentu, yaitu: Home, Exit, SQL, Documentation PHPMyAdmin, dan Dokumentasi MySQL.



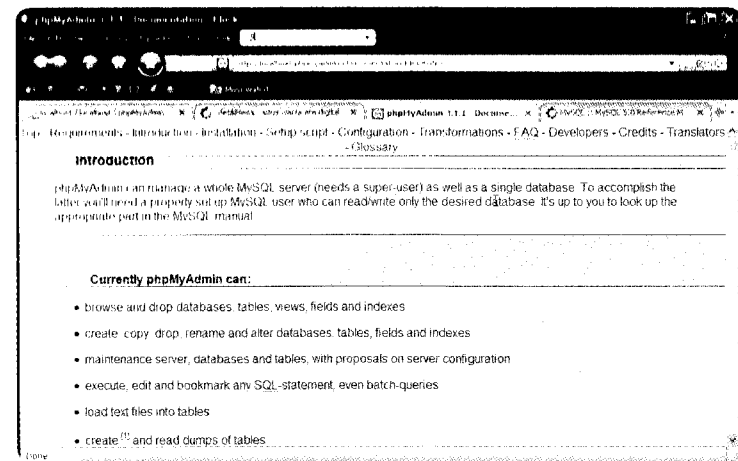
Gambar 5.21 Lima tombol di frame kiri PHPMyAdmin.

1. **Home**, digunakan untuk membuka halaman dasbor/halaman utama PHPMyAdmin.
2. **Exit**, digunakan untuk keluar atau logout dari PHPMyAdmin.
3. **SQL**, akan menampilkan window sql query analyzer yang terpisah (pop up) dan bukan sql query analyzer di halaman PHPMy Admin.



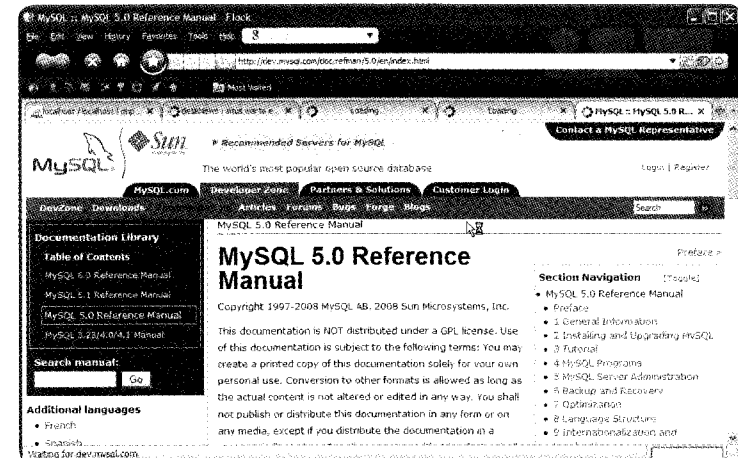
Gambar 5.22 Halaman query analyzer di PHPMyAdmin.

4. **Documentation PHPMyAdmin**, akan menampilkan halaman dokumentasi PHPMyAdmin yang merupakan file-file lokal yang bisa diakses tanpa harus tersambung ke internet.



Gambar 5.23 Dokumentasi PHPMyAdmin.

5. **Dokumentasi MySQL**, menampilkan dokumentasi MySQL yang bisa diakses jika terkoneksi ke internet, karena halaman ini ada di situs resmi MySQL.



Gambar 5.24 Halaman dokumentasi MySQL reference manual di situs dev.MySQL.com.

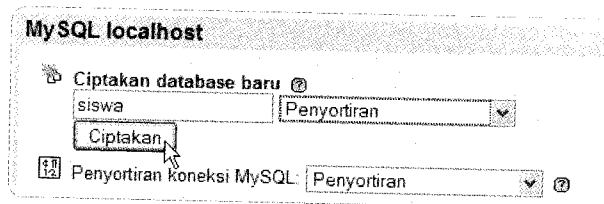
MEMBUAT DAN MENGHAPUS DATABASE

Menggunakan PHPMyAdmin, membuat dan mengolah database dapat dilakukan secara visual sehingga tidak perlu mengetikkan perintah query ataupun membuka MySQL client sama sekali. Karena MySQL client-nya

sudah diganti dengan program PHPMyAdmin ini.

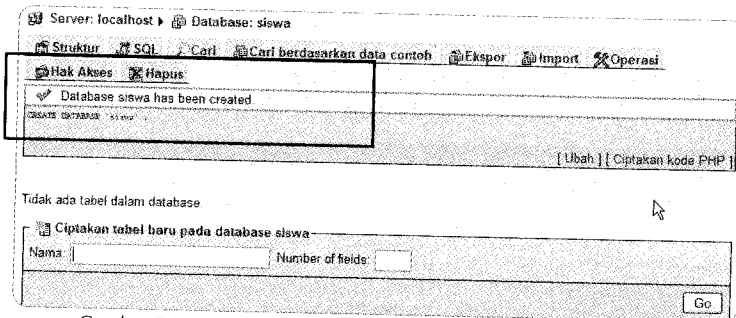
Cara membuat database di PHPMyAdmin versi 3 adalah seperti berikut

1. Dari dasbor, isi nama database yang akan dibuat di textbox **Ciptakan database baru** di kotak **MySQL localhost**. Misalnya, “siswa”, lalu klik **Ciptakan**.



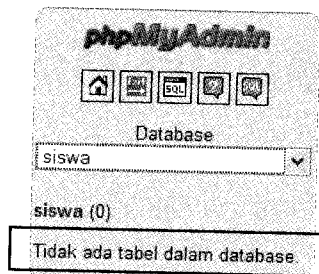
Gambar 5.25 Pengisian nama database di kotak MySQL localhost.

2. Muncul tulisan berikut.



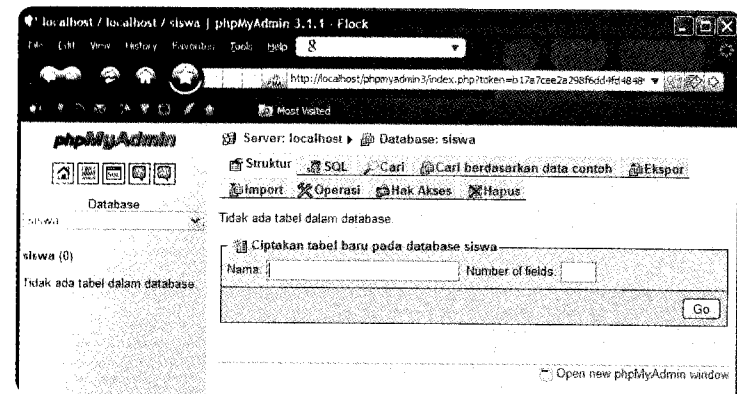
Gambar 5.26 Tampilan PHPMyAdmin setelah database siswa terbuat

Di frame kiri di dasbor PHPMyAdmin, terlihat nama siswa ada di combobox Database. Ini menunjukkan bahwa database sudah terbuat dan sedang dibuka oleh PHPMyAdmin. Karena baru dibuat, belum ada isinya, sehingga ada tulisan:



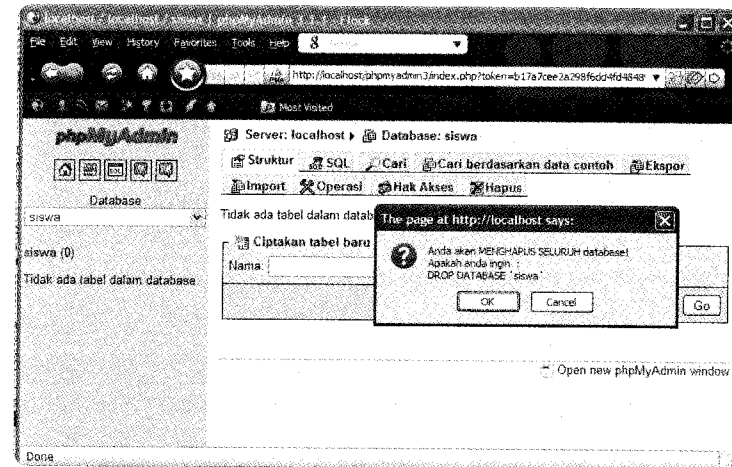
Gambar 5.27 Database siswa sudah terbuat.

Jika nama database tersebut diklik, PHPMyAdmin menampilkan isi database. Interface pengolahan database memiliki 9 tab yang digunakan untuk melakukan pengolahan database.



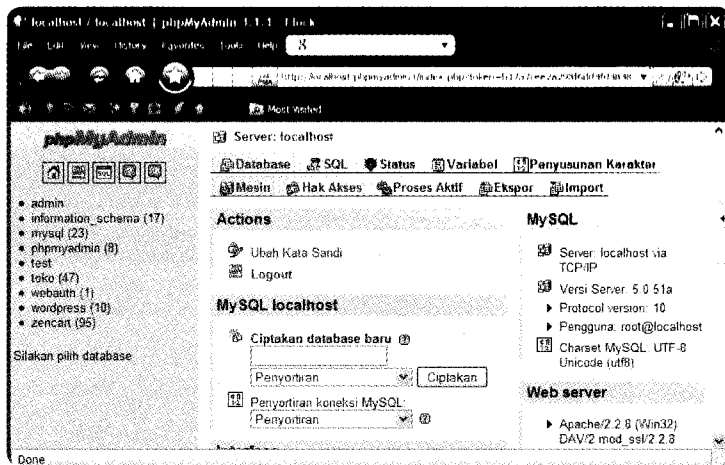
Gambar 5.28 Interface untuk mengolah database di PHPMyAdmin ketika membuka database siswa.

Untuk menghapus database, buka database tersebut kemudian klik tab Hapus. Muncul pertanyaan Anda Akan menghapus seluruh database, apakah Anda ingin drop database 'siswa'? Klik OK jika ingin menghapus database siswa.



Gambar 5.29 Menghapus database siswa.

Setelah database terhapus, nama database tersebut tidak akan terlihat di PHPMyAdmin dan langsung hilang. Database yang sudah terhapus tidak dapat lagi dikembalikan. Karena itu, hati-hatilah dalam menghapus database.

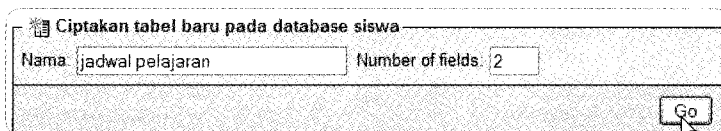


Gambar 5.30 Database siswa sudah dihapus dari MySQL menggunakan PHPMyAdmin versi 3.1.

MEMBUAT TABEL

Begitu pun ketika membuat tabel, dengan PHPMyAdmin membuat tabel tidak perlu mengetik secara manual. Dengan demikian, PHPMyAdmin ini memang sangat cocok dipakai untuk mengadministrasi database MySQL oleh admin yang tidak ingin kerepotan. Cara membuat tabel adalah:

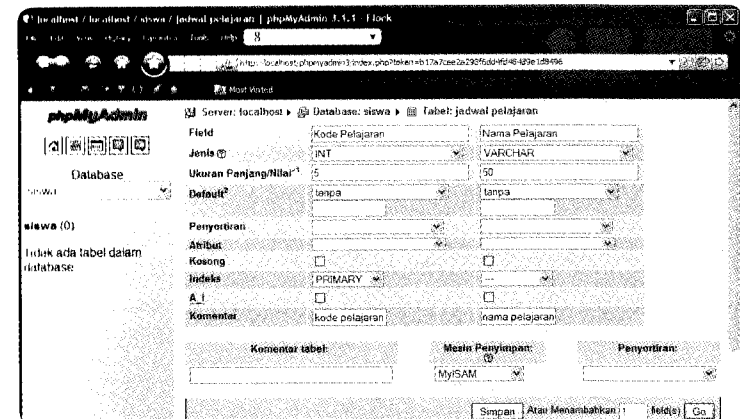
1. Buka database yang ingin dibuatkan tabel di dalamnya, misalnya database siswa. Lalu isi nama tabel yang akan dibuat, misalnya 'jadwal pelajaran' di textbox **Nama**. Jumlah kolom dari tabel diisikan di kotak **Number of fields**. Klik **Go** untuk membuat tabel tersebut.



Gambar 5.31 Pembuatan tabel 'jadwal pelajaran' dengan 2 field di tabel siswa.

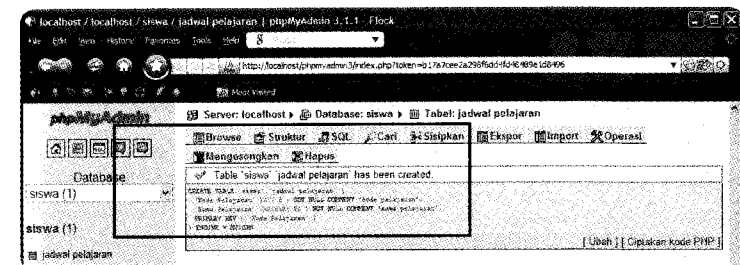
2. Isi nama 2 field untuk tabel di bagian **Field**, misalnya **Kode Pelajaran** dan **Nama Pelajaran**. Jenis field diisikan di **Jenis**, yaitu integer dan varchar. Ukuran maksimal diisikan di **Ukuran panjang/nilai**. Jika diperlukan, isi jenis indeks dan komentar untuk field tersebut. Klik

Simpan untuk menyimpan field tersebut, atau klik isikan jumlah field tambahan yang akan ditambahkan kemudian klik **Go** untuk menambahkan field baru.



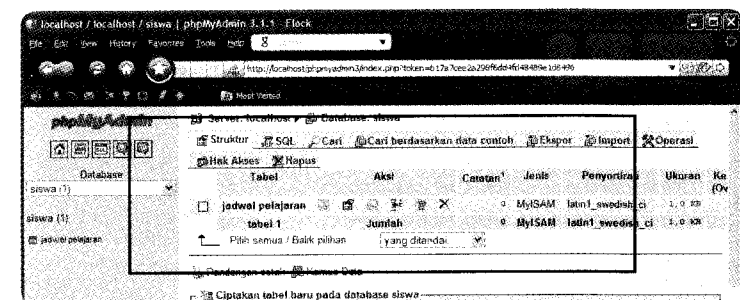
Gambar 5.32 Menambahkan field baru.

3. Tabel jadwal pelajaran sudah terbuat di database siswa.



Gambar 5.33 Tabel jadwal pelajaran dengan 2 field sudah terbuat di database siswa.

ketika database siswa dibuka di PHPMyAdmin, tabel yang dibuat terlihat di frame sebelah kiri dan juga frame sebelah kanan.

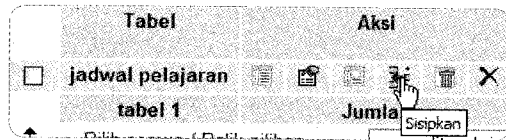


Gambar 5.34 Keterangan di halaman Database siswa bahwa tabel 'jadwal pelajaran' sudah terbuat.

MENGISI DAN MENGUBAH DATA DI TABEL

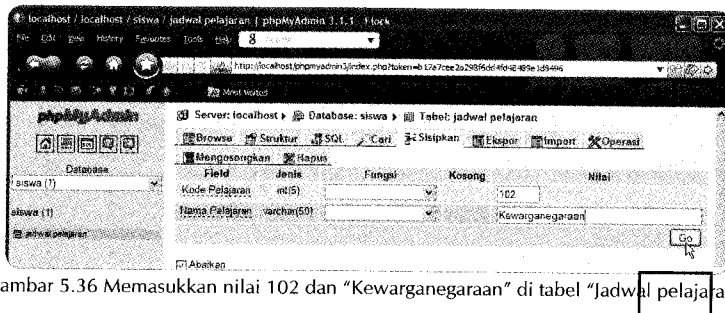
Tabel adalah inti dari database, termasuk juga di MySQL. Untuk mengisi dan mengubah tabel, caranya sebagai berikut:

1. Buka halaman database siswa di PHPMYAdmin.
2. Di bagian tabel 'jadwal pelajaran', klik tombol **Sisipkan** di kolom Aksi.



Gambar 5.35 Klik pada tombol Sisipkan di samping kanan.

3. Isi nilai untuk masing-masing pada kolom Nilai. Kemudian klik **Go**.

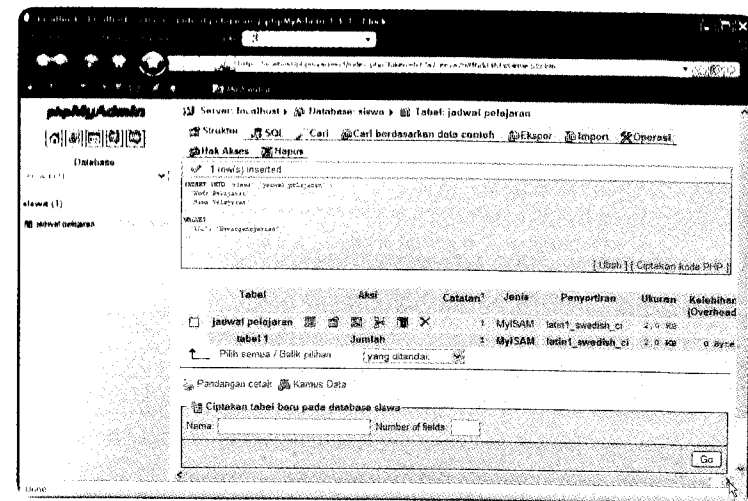


Gambar 5.36 Memasukkan nilai 102 dan "Kewarganegaraan" di tabel "Jadwal pelajaran".

Setelah diklik **Go**, muncul pemberitahuan bahwa satu record baris di tabel sudah dimasukkan.

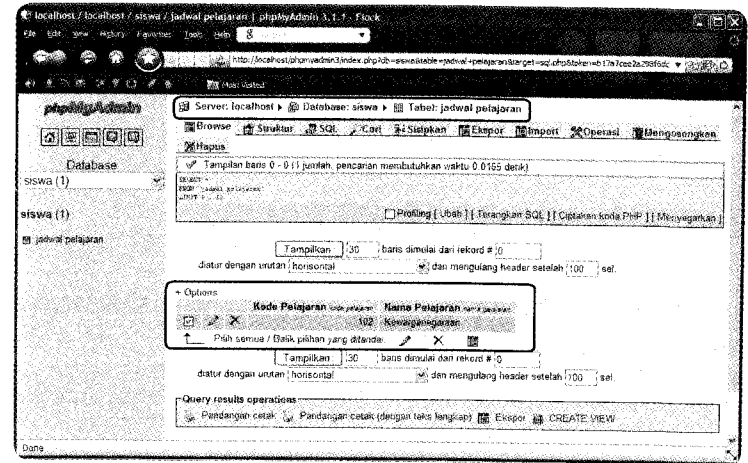
Di bawahnya terdapat kode sql statement yang dieksekusi untuk menambahkan data ke dalam MySQL. Kode ini nantinya bisa di copy paste untuk membuat sql statement di PHP.

```
INSERT INTO `siswa`.`jadwal pelajaran` (`Kode Pelajaran`,  
`Nama Pelajaran`)VALUES (  
'102', 'Kewarganegaraan');
```



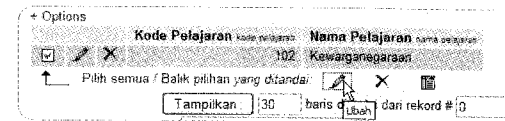
Gambar 5.37 Tampilan halaman tabel 'jadwal pelajaran' PHPMYAdmin setelah memasukkan record.

Untuk melihat nilai data yang sudah dimasukkan, klik pada nama tabel di menu breadcrumb di atas, kemudian klik tab **Browse**.



Gambar 5.38 Klik browse.

Untuk mengganti nilai di tabel, ceklis pada baris yang ingin diubah, lalu klik pada tombol **Ubah** di bawah tabel.



Gambar 5.39 Klik pada tombol Ubah di bawah tabel setelah sebelumnya memberi cek pada baris yang akan diubah.

Ganti teks yang ingin diubah, misalnya Kewarganegaraan ingin diganti 'PPKn'.

Field	Jenis	Fungsi	Kosong	Nilai
Kode Pelajaran	int(5)			102
Nama Pelajaran	varchar(50)			PPKn

Gambar 5.40 Perubahan nilai baris di tabel

Ada pemberitahuan beserta kode sql statement yang digunakan untuk mengubah baris tertentu tersebut.

```
UPDATE `siswa`.`jadwal pelajaran` SET
`Nama Pelajaran` = 'PPKn' WHERE `jadwal
pelajaran`.`Kode Pelajaran` =102 LIMIT 1 ;
```

Hasilnya, seperti gambar berikut.

Kode Pelajaran	Nama Pelajaran
102	PPKn

Gambar 5.41 Perubahan isi baris tabel

MENCARI NILAI TERTENTU

Mencari nilai tertentu di PHPMyAdmin dapat dilakukan dengan menggunakan cara berikut.

1. Cari tahu semua isi tabel, dengan mengklik tab Browse. Tereksekusi secara otomatis perintah sql:

```
SELECT *
FROM `jadwal pelajaran`
LIMIT 0 , 30
```

2. Semua isi record di tabel jadwal pelajaran terlihat sebagai berikut.

Kode Pelajaran	Nama Pelajaran
102	PPKn
101	Matematika Dasar
100	Matematika Teknik
105	Termodinamika
106	Mekanika Teknik

Gambar 5.42 Semua record di tabel jadwal pelajaran dibuka.

1. Untuk mencari record tertentu, klik nama tabel di menu bread crumb lalu klik tab **Cari**.

Gambar 5.43 Klik pada tab Cari untuk mencari isi tabel jadwal pelajaran.

1. Ketika mencari dengan parameter tertentu, ada beberapa operator yang perlu diketahui jenis dan deskripsinya. Berikut ini daftar operator yang ada di MySQL:

Tabel 5.1 Daftar operator di MySQL

Nama	Deskripsi
AND, &&	Logical AND
BETWEEN ... AND ...	Check whether a value is within a range of values
BINARY	Cast a string to a binary string
&	Bitwise AND
~	Invert bits
	Bitwise OR
^	Bitwise XOR
CASE	Case operator
DIV(v4.1.0)	Integer division
/	Division operator
<=>	NULL-safe equal to operator

=	Equal operator
>=	Greater than or equal operator
>	Greater than operator
IS NOT NULL	NOT NULL value test
IS NOT	Test a value against a boolean
IS NULL	NULL value test
IS	Test a value against a boolean
<<	Left shift
<=	Less than or equal operator
<	Less than operator
LIKE	Simple pattern matching
-	Minus operator
%	Modulo operator
NOT BETWEEN ...	Check whether a value is not AND ... within a range of values
!=, <>	Not equal operator
NOT LIKE	Negation of simple pattern matching
NOT REGEXP	Negation of REGEXP
NOT, !	Negates value
, OR	Logical OR
+	Addition operator
REGEXP	Pattern matching using regular expressions
>>	Right shift
RLIKE	Synonym for REGEXP
SOUNDS LIKE(v4.1.0)	Compare sounds

+	Times operator
-	Change the sign of the argument
XOR	Logical XOR

Lalu, isikan operator di kolom operator dan masukkan operand yang digunakan sebagai parameter pencarian di kolom Nilai. Ketika mengisi kolom nilai tersebut, ada tampilan drop down yang meng-enumerasi daftar nama yang ada di kolom pada tabel tersebut sehingga memudahkan Anda mengisi.

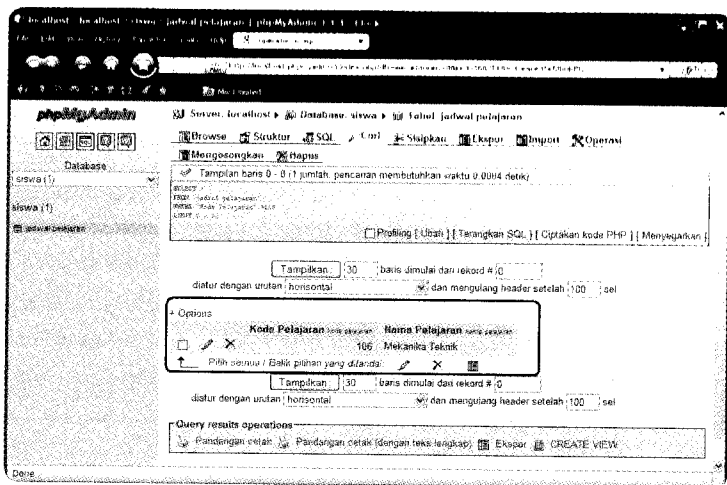
Gambar 5.44 Pengisian operand di kolom Nilai.

- Klik **Go**. PHPMyAdmin akan mengeksekusi sql query secara otomatis. Misalnya, seperti berikut:

Tampilan baris 0-0 (1 jumlah, pencarian membutuhkan waktu 0.0004 detik)

```
SELECT *
FROM `jadwal_pelajaran`
WHERE `Kode Pelajaran` =106
LIMIT 0 , 30
```

Tampilan record yang sesuai dengan operand yang diisikan terlihat di halaman **Cari** seperti berikut.

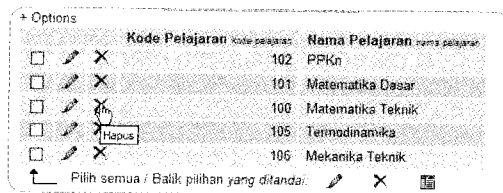


Gambar 5.45 Pencarian halaman cari.

MENGHAPUS RECORD

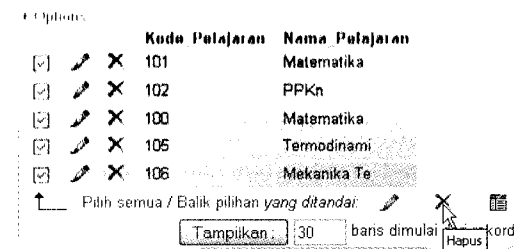
Menghapus record di tabel dapat dilakukan secara visual di PHPMyAdmin, dengan cara seperti berikut:

1. Buka semua record dengan mengklik tab Browse. Misalnya, ada beberapa nilai seperti berikut:



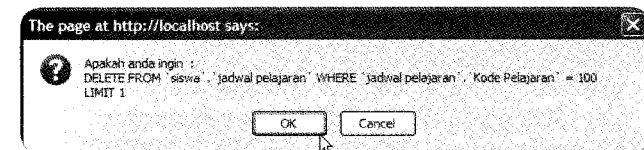
Gambar 5.46 Semua nilai di tabel.

2. Pilih baris record yang ingin dihapus kemudian klik tombol hapus (tanda silang) di bagian baris tersebut. Jika ingin menghapus banyak record sekaligus, ceklis baris-baris yang akan dihapus, kemudian klik tombol Hapus (tanda silang) di bagian bawah.



Gambar 5.47 Menghapus banyak record.

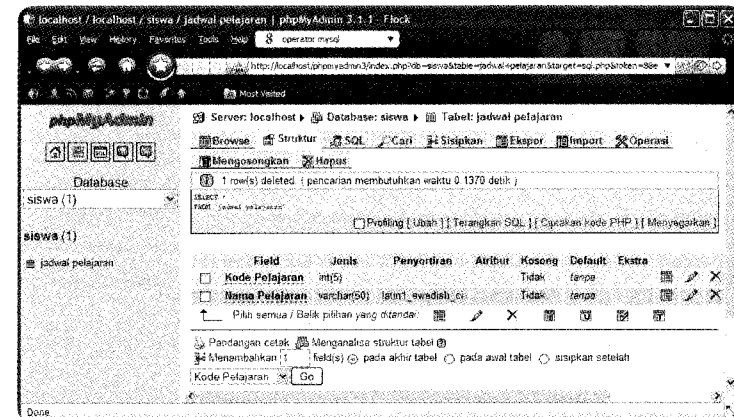
3. Klik OK untuk menghapusnya.



Gambar 5.48 Konfirmasi penghapusan satu record.

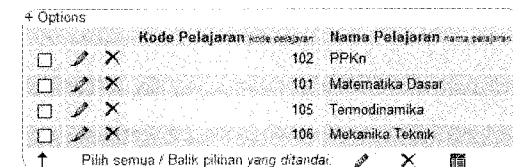
PHPMyAdmin mengeksekusi sql query penghapusan secara otomatis.

1 row(s) deleted. (pencarian membutuhkan waktu 0.1370 detik)
SELECT * FROM `jadwal\_pelajaran`



Gambar 5.49 Tampilan PHPMyAdmin setelah menghapus satu record.

Record yang terhapus hilang dari tabel ketika dibuka dengan mengklik tab **Browse**.



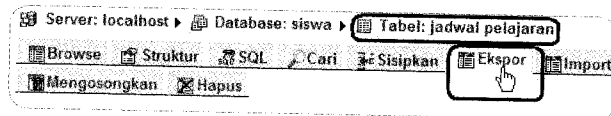
Gambar 5.50 Tampilan isi tabel jadwal pelajaran setelah satu record dengan kode 100 dihapus.

EKSPOR DAN IMPOR DATA

Pada PHPMyAdmin, Anda bisa mengekspor database. Ekspor data adalah memindahkan data ke dalam format sql atau format lainnya sehingga nantinya mudah diimpor ke tempat lain. Jadi, dengan ekspor, backup sangat praktis nantinya.

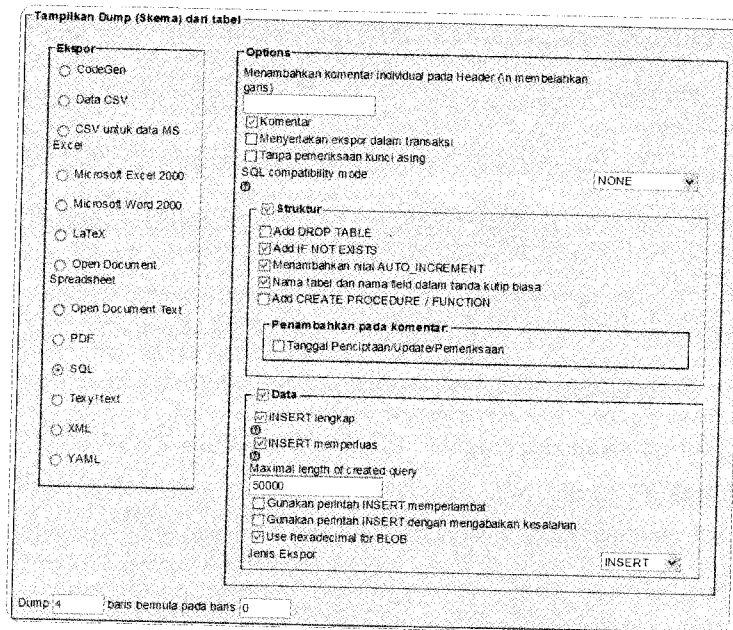
Ekspor data dapat ditentukan dalam level database ataupun tabel. Misalnya, ingin membackup tabel, caranya sebagai berikut.

1. Klik pada menu tabel di breadcrumb kemudian klik tab **Ekspor**.



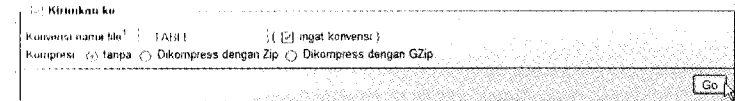
Gambar 5.51 Klik Ekspor untuk membackup tabel.

2. Sebagai contoh, akan diekpor dalam bentuk SQL, dimana nantinya akan menghasilkan kode sql plain text.



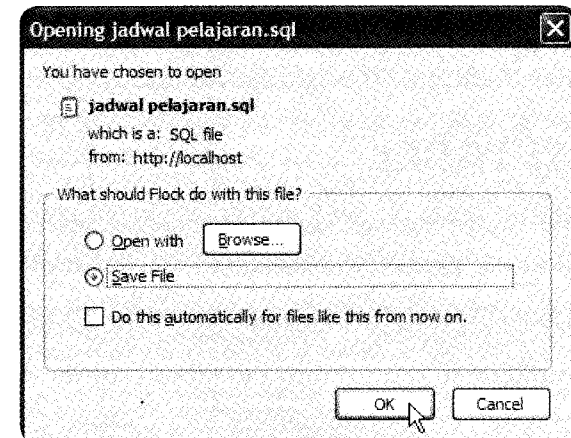
Gambar 5.52 Pemilihan jenis ekspor di tab Ekspor.

3. Jika Anda ingin menyimpan file ekspor dalam komputer lokal, beri ceklis pada **Kirimkan ke** baru klik **Go**.



Gambar 5.53 Kirimkan ke untuk menyimpan file ke ekstensi .sql.

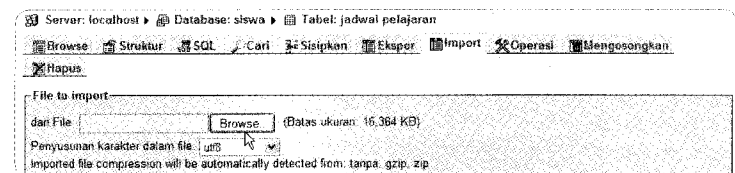
4. Akan muncul kotak dialog seperti gambar di bawah. Klik **Ok** untuk mendownload dan menyimpannya di komputer lokal.



Gambar 5.54 Download file jadwal pelajaran.sql yang merupakan hasil ekspor dari isi di tabel jadwal pelajaran.sql.

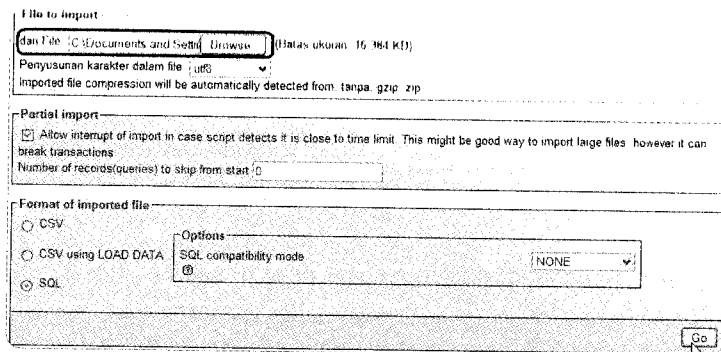
Impor ini merupakan proses kebalikan dari ekspor. Cara mengimpor data adalah sebagai berikut:

1. Klik tab **Import**.
2. Kemudian klik tombol **Browse** di kotak **File to import**.



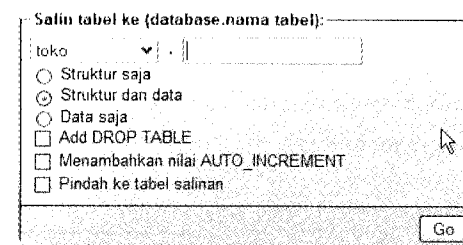
Gambar 5.55 Halaman Import untuk mengimpor data di PHPMyAdmin.

3. Klik **Go**.



Gambar 5.56 Proses impor di PHPMyAdmin.

Tabel bisa disalin ke tempat yang lain dengan menggantinya di **Salin tabel ke**. Tidak hanya bisa menyalin ke database yang sama, tetapi juga di database yang berbeda, asalkan masih di server MySQL yang sama.

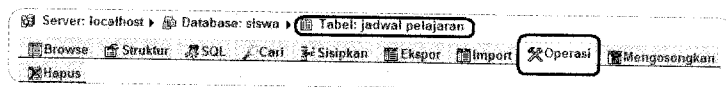


Gambar 5.59 Penyalinan di Salin tabel ke.

OPERASI TABEL

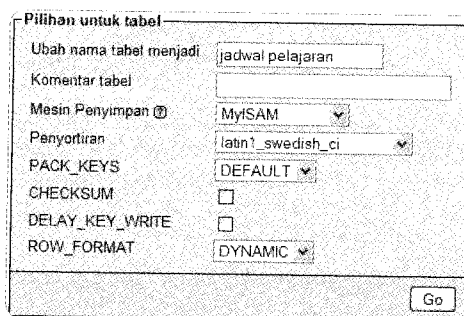
Tabel dapat diubah strukturnya maupun dipindahkan atau disalin ke tabel lain di database. Cara melakukan operasi tabel adalah:

1. Klik nama tabel di menu breadcrumb. Kemudian klik tab **Operasi**.



Gambar 5.57 Pemilihan tabel.

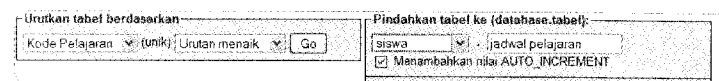
2. Untuk mengubah operasi pilihan tabel, seorang bisa mengubah nilai di kotak **Pilihan untuk tabel**, semisal mengubah nama tabel.



Gambar 5.58 Pengubahan pilihan untuk tabel.

Tabel juga bisa diganti pengurutannya (sorting) berdasarkan parameter tertentu dan juga dengan aturan tertentu (ascending atau descending) di bagian **Urutkan tabel berdasar**.

Tabel pun dapat dipindahkan. Jika dipindahkan, akan menghapus tabel yang lama. Jadi, tidak seperti tabel yang sudah ada.

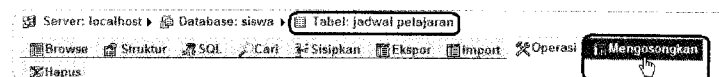


Gambar 5.60 Pengurutan dan pemindahan tabel.

TRUNCATE DAN DROP

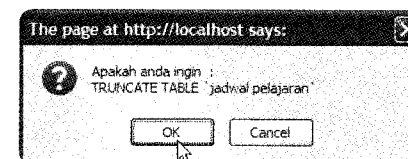
Truncate untuk mengosongkan semua isi tabel. Caranya dengan:

1. Klik nama tabel, lalu klik tab **Mengosongkan**.



Gambar 5.61 Tab untuk mengosongkan tabel dengan truncate.

2. Klik **OK** untuk mengeksekusinya.



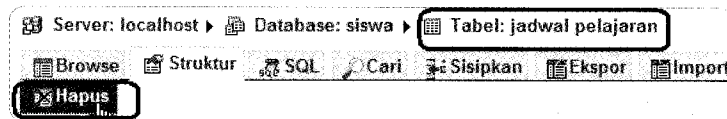
Gambar 5.62 Konfirmasi truncate tabel 'jadwal pelajaran'.

Sql statement untuk truncate pun tereksekusi dengan otomatis.

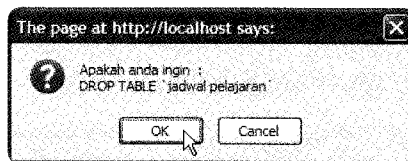
Tabel jadwal pelajaran telah dikosongkan (pencarian membutuhkan waktu 0.0016 detik)

```
TRUNCATE TABLE `jadwal_pelajaran`
```

Sementara Drop akan menghapus struktur tabel beserta isinya. Caranya dengan memilih tabel lalu klik tab Hapus. Begitu ada konfirmasi, langsung klik **OK**.



Gambar 5.63 Pemilihan tab Hapus untuk men-drop tabel.



Gambar 5.64 Konfirmasi drop tabel.

ODBC

Bab ini akan membahas:

- Instalasi ODBC data adapter.
- Pengaksesan MySQL dengan Acces 2007.
- Pengaksesan MySQL dengan Open Office Base 3.

Odbc merupakan sebuah konektor yang fungsinya menghubungkan/koneksi database menggunakan API (application programming interface) ODBC di semua platform Microsoft Windows dan Unix. Konsekuensinya, database dapat diakses dengan mudah di lingkungan pemrograman tertentu, seperti Microsoft Access, Excel, ataupun pemrograman lain yang mendukung ODBC seperti Visual Basic dan Delphi.

Ada 2 fitur utama odbc untuk MySQL, yaitu:

1. Sesuai dengan standar (Standards compliant): Versi ODBC Connector/ODBC sesuai dengan standar odbc 2.50 level 0 di mana beberapa di antaranya sesuai dengan level 1 dan 2. Versi stabilnya juga memenuhi persyaratan odbc 3.51 level 1 dan level 2.
2. Multiplatform, Connector/ODBC untuk MySQL tersedia untuk Microsoft Windows, dan juga berbagai sistem Unix (termasuk Linux, Sun Solaris, HP-UX, IBM AIX, dan Mac OS X).

Ada 2 versi MySQL Connector/ODBC yang tersedia untuk didownload dari situs odbc MySQL, yaitu di <http://www.mysql.com/products/connector/odbc/>:

- MySQL Connector/ODBC 3.51, yaitu versi yang siap digunakan untuk produksi dan tersedia dalam bentuk lisensi GPL.
- MySQL Connector/ODBC 2.50, yaitu versi lama yang tersedia dalam bentuk lisensi LGPL.

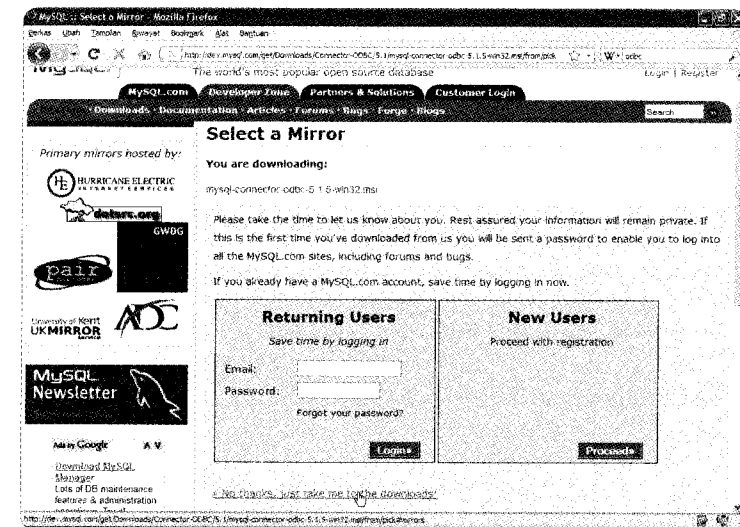
Jadi, fungsi utama Open Database Connectivity (ODBC) adalah menyediakan API Standar untuk penggunaan software database management systems (DBMS). Implikasinya apa? Kegiatan manajemen database menjadi lebih mudah karena proses manajemen database menjadi tidak tergantung kepada bahasa pemrograman, sistem database, dan sistem operasi.

Beberapa sistem operasi yang didukung oleh connector ODBC/MySQL adalah:

- Windows
- Windows x64

- Linux (non RPM packages)
- Linux x86 generic RPM (dynamically linked)
- Linux AMD64 / Intel EM64T generic RPM
- Linux Intel IA64 generic RPM
- Solaris (pkgadd packages)
- Solaris (TAR packages)
- FreeBSD (TAR packages)
- Mac OS X (package format)
- Mac OS X (TAR packages)
- HP-UX (TAR packages)
- IBM AIX
- Source

Ketika akan mendownload MySQL odbc connector. Pengunjung bisa memilih apakah akan login dahulu sebagai user atau tidak. Untuk cepatnya, lebih baik tak usah login, klik link **No thanks, just take me to the downloads**.



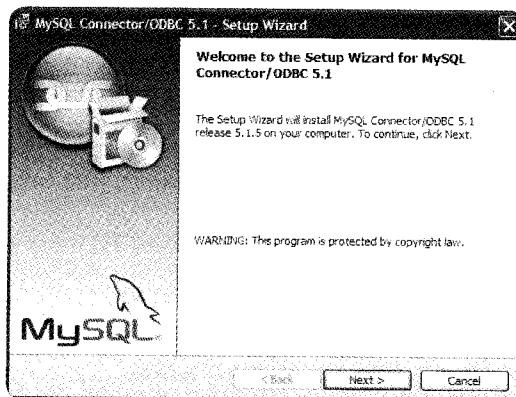
Gambar 6.1 Halaman download ODBC Connector MySQL.

Pilih server mirror dari Indonesia, yaitu dari [Cyberindo Aditama], klik link HTTP atau FTP. Versi terbaru saat buku ini ditulis adalah 5.1.5. Namun, Anda dapat juga menggunakan versi yang lebih mutakhir.

INSTALASI ODBC MYSQL CONNECTOR 5.1.5

Odbc driver untuk MySQL harus diinstal terlebih dahulu di komputer sebelum aplikasi-aplikasi tertentu di komputer dapat mengakses data di MySQL. Cara instalasinya adalah:

1. Klik dua kali installer lalu klik **Next**.

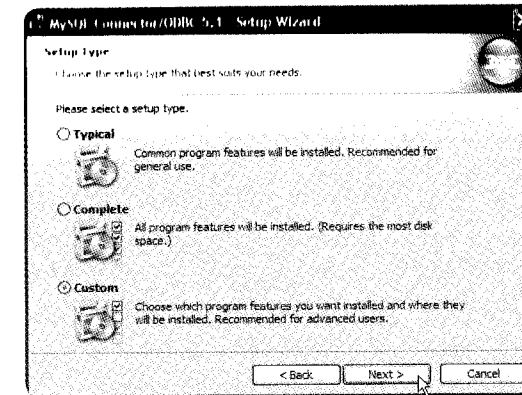


Gambar 6.2 Halaman pertama MySQL connector/odbc setup wizard.

Ada 3 pilihan instalasi MySQL connector/odbc, yaitu:

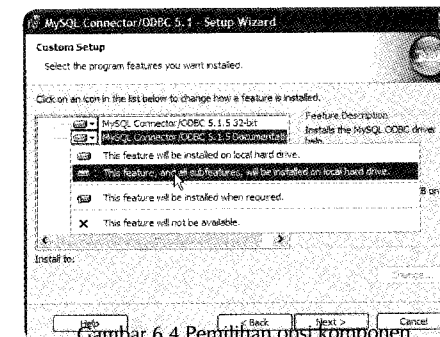
- **Typical**, yaitu instalasi yang disarankan untuk pemula.
- **Complete**, yaitu instalasi semua bagian.
- **Custom**, yaitu instalasi bagian tertentu.

2. Sebagai contoh, pilih **Custom** lalu klik **Next**.



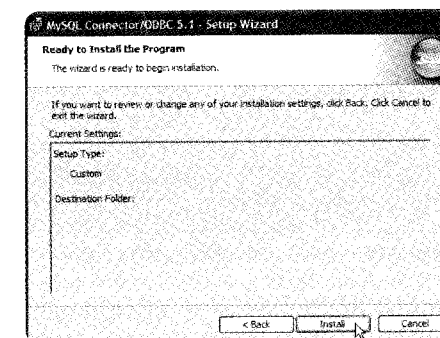
Gambar 6.3 Penentuan setup type ke custom.

3. Instalasi **Custom** memungkinkan memilih apakah sebuah komponen diinstalasi di harddisk lokal, baik fitur atau sub fiturnya, atau semuanya. Atau, menentukan apakah tidak diinstal sama sekali.



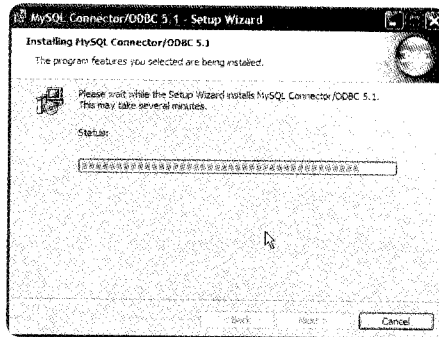
Gambar 6.4 Pemilihan opsi komponen

4. Klik **Install**.



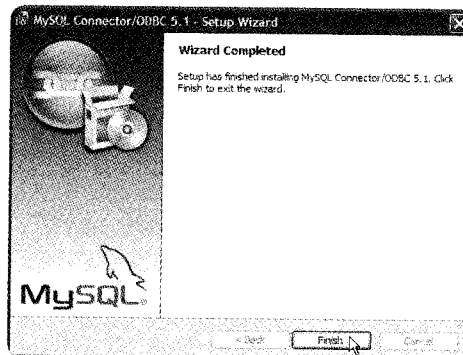
Gambar 6.5 Instalasi custom.

Ketika proses instalasi dimulai, Anda dapat melihat status instalasi di progress bar. Jangan klik **Cancel** karena akan membatalkan proses instalasi.



Gambar 6.6 Proses instalasi MySQL connector/odbc 5.1 sedang dilaksanakan.

5. Klik **Finish**.



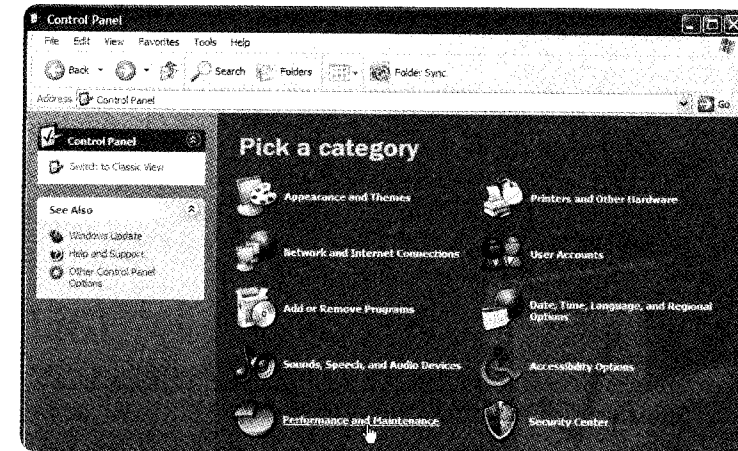
Gambar 6.7 Instalasi MySQL connector/odbc 5.1 selesai.

MEMBUKA ODBC DATA SOURCE ADMINISTRATOR

Ketika MySQL ODBC Connector 5.1.5 sudah terinstal, berarti Windows siap menggunakannya. Artinya, Windows dan software-software yang bisa mengakses library odbc bisa mengakses database secara langsung.

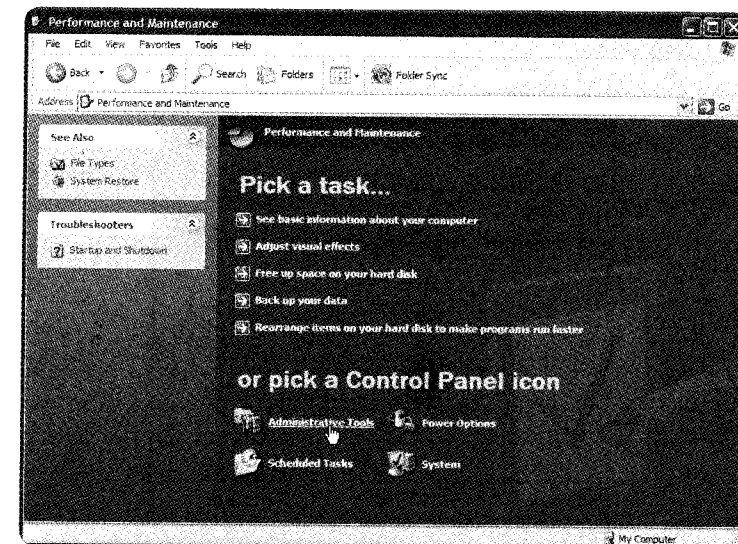
Agar dapat mengakses ODBC, Anda harus terlebih dahulu membuka halaman ODBC data source administrator dengan cara seperti berikut:

1. Buka **Control Panel** di Windows dengan meng-klik **Start > Control Panel** pada **Start** menu Windows.
2. Klik ikon **Performance and Maintenance**.



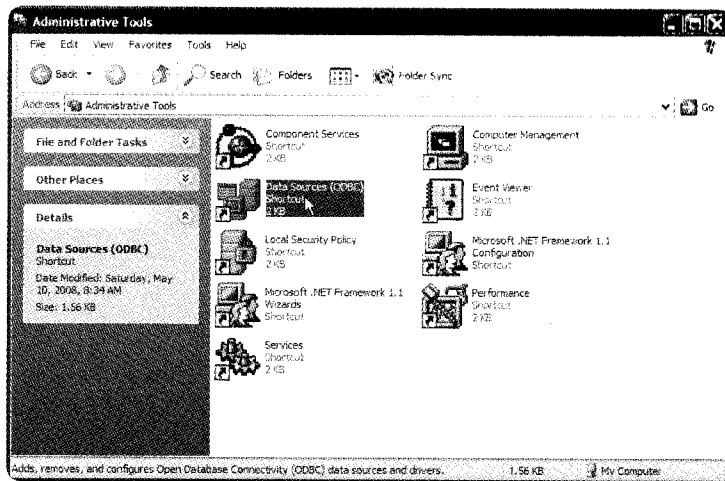
Gambar 6.8 Kategori Performance and Maintenance yang ada di Control Panel.

3. Klik ikon **Administrative Tools**.



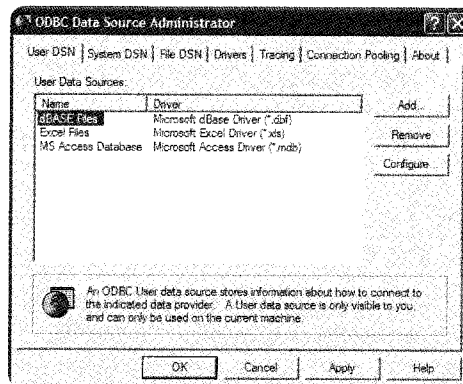
Gambar 6.9 Pemilihan ikon Administrative Tools.

4. Klik ikon Data Sources (ODBC) Shortcut.



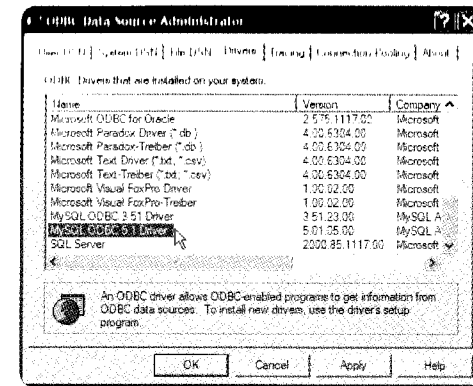
Gambar 6.10 ODBC Data Sources di window Administrative Tools.

Window ODBC Data Source Administrator terbuka. Ini adalah jendela yang berfungsi untuk mengakomodasi administrasi data source odbc di mana Anda bisa menambah, menghapus, atau mengonfigurasi data source untuk menghubungkan diri ke berbagai sumber data yang didukung oleh odbc.



Gambar 6.11 ODBC Data Source Administrator.

Sementara tab **Drivers** menunjukkan adanya indikasi bahwa odbc MySQL connector sudah terbuat.



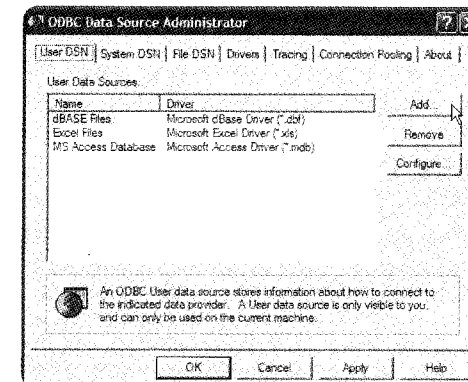
Gambar 6.12 Indikasi adanya driver odbc untuk mysql di tab Drivers.

MEMBUAT USER DSN

Data sources information (DSN) adalah sebuah informasi sumber data yang dapat digunakan di Windows. Sebelum bisa digunakan, dsn harus diidentifikasi terlebih dahulu dengan cara membuatnya di bagian ODBC Data Source Administrator.

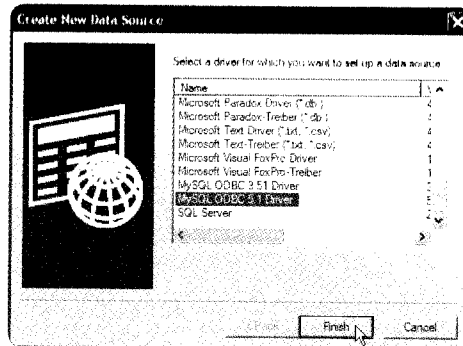
Cara untuk membuat user dsn adalah:

1. Masuk ke halaman ODBC data source administrator.
2. Klik tab **User dsn**.
3. Klik **Add** untuk menambahkan user dsn untuk MySQL.



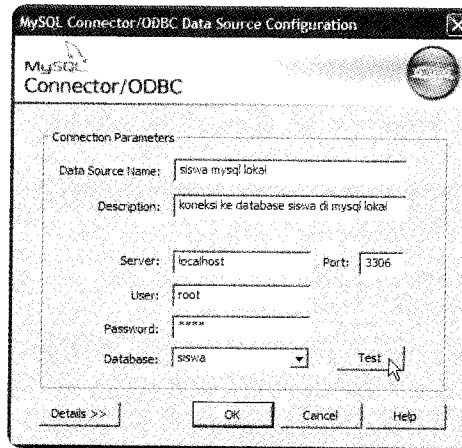
Gambar 6.13 Klik pada tombol Add untuk menambah user dsn baru.

- Pilih **MySQL odbc 5 Driver** lalu klik **Finish**.



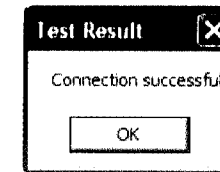
Gambar 6.14 Memilih driver untuk DSN baru.

- Isi nama data source di kotak teks **Data Source Name**.
- Deskripsi data source diisikan di bagian **Description**.
- Untuk server, user, dan password sesuai dengan lokasi dan user MySQL. Untuk lokal, isi localhost.
- Pilih database di bagian kotak combo **Database** kemudian klik **Test**.



Gambar 6.15 Pengisian parameter koneksi.

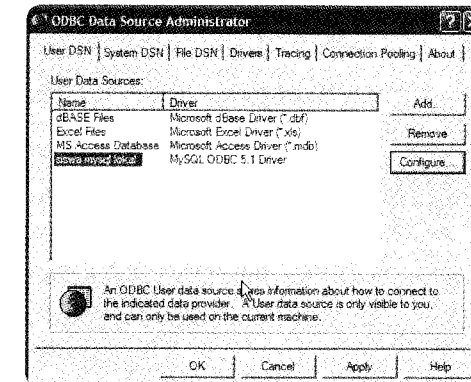
Jika koneksi benar, dan tidak ada masalah, muncul pemberitahuan **Test Result** yang menyatakan bahwa koneksi sudah sukses dibuat.



Gambar 6.16 Koneksi sudah sukses dibuat.

- Klik **OK** dua kali.

Tampilan nama dsn user akan terlihat di bagian kotak **User Data Sources**. Koneksi dsn ini nantinya siap dipakai di aplikasi-aplikasi lain yang memerlukan.



Gambar 6.17 DSN user ke database siswa di server MySQL sudah terkoneksi.

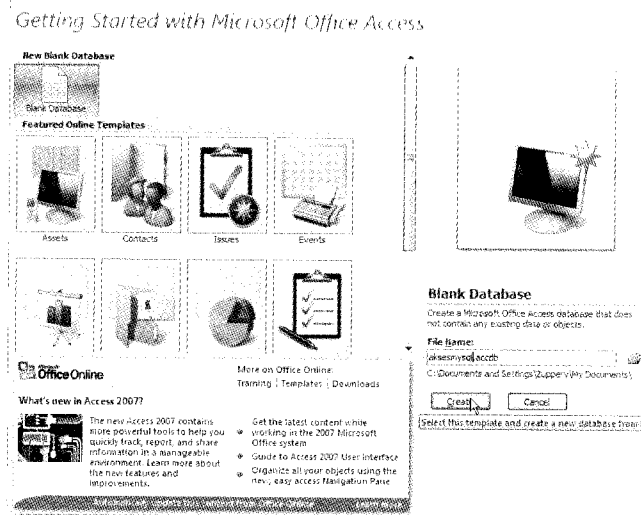
MENGHUBUNGKAN MICROSOFT ACCESS MYSQL

Microsoft access dapat digunakan untuk memodifikasi ataupun mengambil data dengan metode link dari MySQL. Caranya pun sangat mudah, yaitu memanfaatkan user dsn yang sudah ada. Jika belum ada, Anda bisa membuatnya secara manual.

Berikut ini contoh cara menghubungkan Microsoft Access ke MySQL:

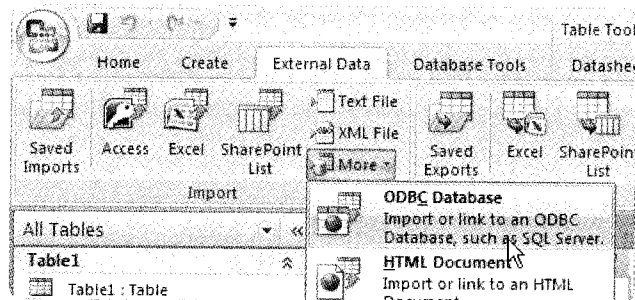
- Buka Microsoft Access kemudian buat sebuah blank database, lalu isi nama database di **File name**. Di MS access, nama database adalah nama file ditambah ekstensi .accdb.

2. Klik **Create** untuk membuat database baru.



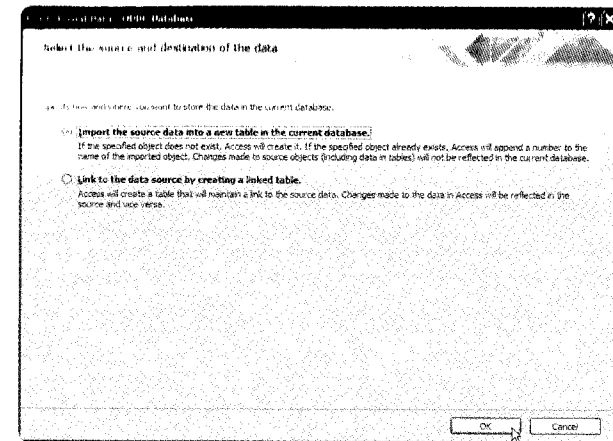
Gambar 6.18 Pembuatan blank database.

Database Acces yang pertama kali Anda buat belum mempunyai tabel. Akan tetapi, jika Anda sudah mempunyai sumber data eksternal, Anda dapat memanfaatkannya untuk digunakan dalam Microsoft Acces. Caranya dengan mengklik tab **External Data** di **Ribbon** kemudian mengklik button **More > ODBC Database**.



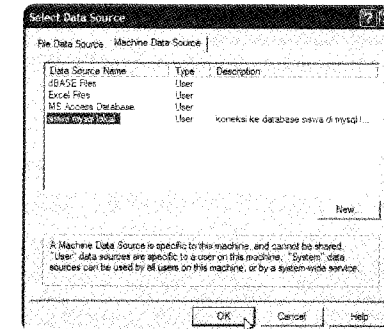
Gambar 6.19 Button More > ODBC database di bagian tab External data.

Pengambilan data via odbc di access dapat melalui beberapa tahapan. Pertama, mengimpor source data ke tabel baru, sementara yang kedua dengan me-link. Pilihlah caranya di bagian **Specify how and where you want to store the data in the current database**.



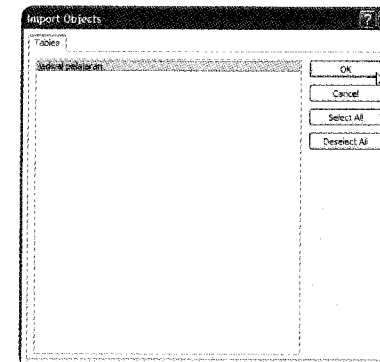
Gambar 6.20 Pemilihan opsi impor data dari odbc ke MS Access.

1. Klik **OK**.
2. Klik pada data source yang ada kemudian klik **OK**.



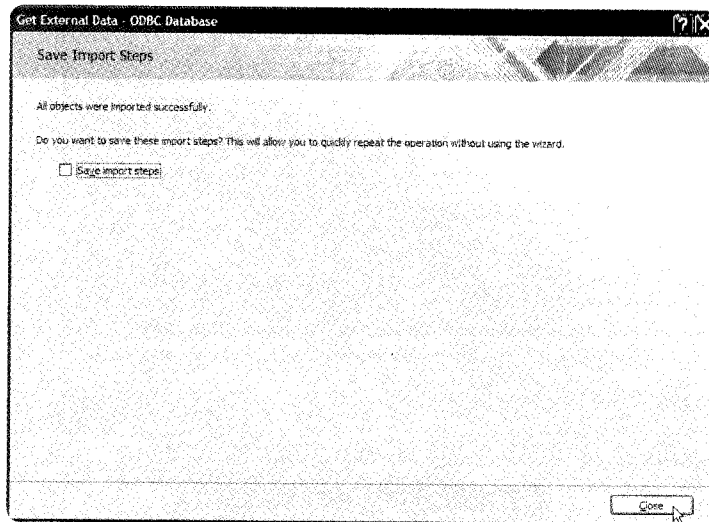
Gambar 6.21 Pemilihan user dsn untuk sumber data eksternal.

3. Pilih tabel yang ada di odbc tersebut, lalu klik **OK**. Tabel yang terpilih nantinya akan diimpor datanya ke MS Access.



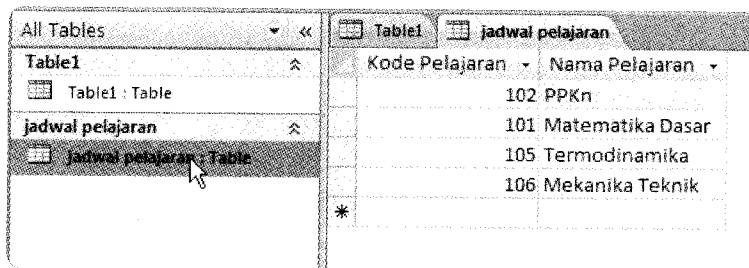
Gambar 6.22 Pemilihan tabel yang akan diimpor ke dalam Access.

4. Apabila ingin menyimpan langkah langkah impor, ceklis **Save import steps**, kemudian klik **Close**.



Gambar 6.23 Permintaan untuk menyimpan Save import steps.

Tabel baru otomatis terbuat, terlihat di frame **All Tables**. Jika nama tabel tersebut diklik, di dalamnya sudah terdapat beberapa record sesuai dengan yang telah diisi di database MySQL yang asli.



Gambar 6.24 Tabel baru terbuat di MS Access dengan sumber data dari MySQL melalui odbc.

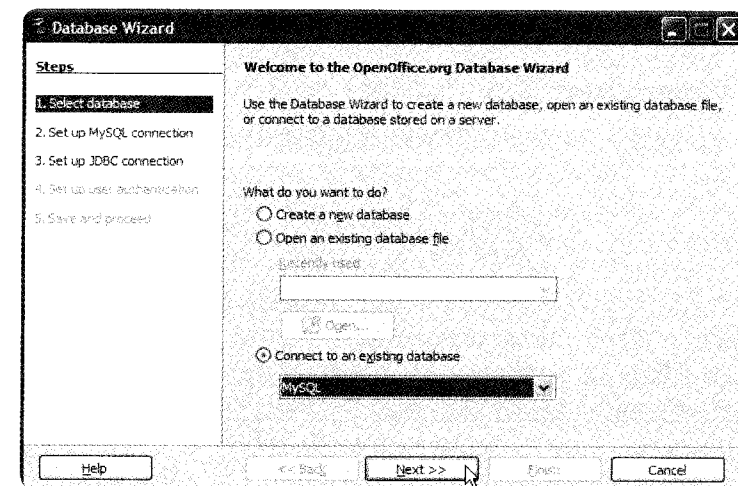
OPENOFFICE.ORG BASE 3 DENGAN MYSQL

Versi open office terbaru ketika buku ini ditulis adalah open office 3. Dukungan OpenOffice.org 3 ke MySQL lebih baik dibandingkan access karena bisa langsung menjadi front end dari database MySQL. Anda dapat menggunakan OpenOffice.org base sebagai user interface untuk pengolahan data di database MySQL sehingga mempermudah proses manajemen database.

Ini mengingat kecenderungan bahwa dukungan sesama software open source biasanya lebih baik dibandingkan dukungan software komersial terhadap open source.

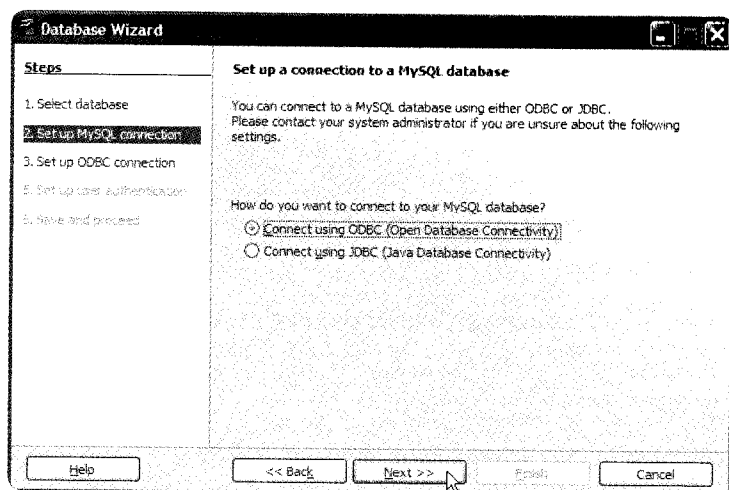
Berikut ini cara menggunakan OpenOffice.org base 3 sebagai user interface untuk MySQL:

1. Buka OpenOffice.org base 3.
2. Pilih **connect to an existing database**.
3. Pilih item **MySQL** di kotak combo **Connect to an existing database**. Kemudian klik **Next**.



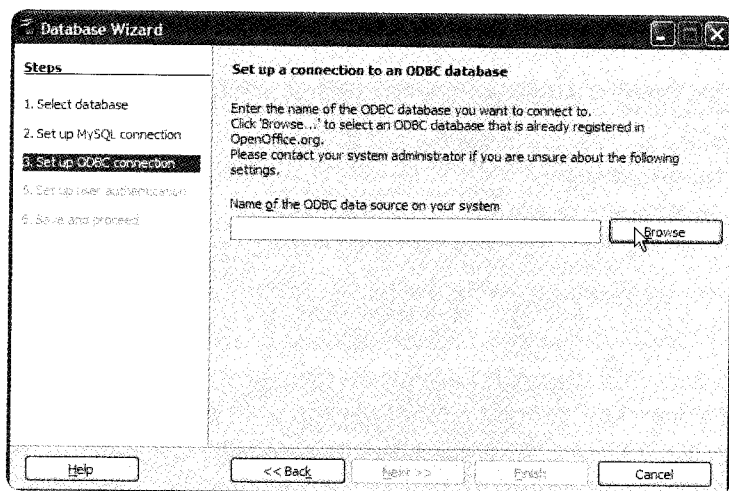
Gambar 6.25 Memilih open office base 3 agar terhubung ke database MySQL.

4. Klik **Connect using ODBC (Open Database Connectivity)** lalu klik **Next**.



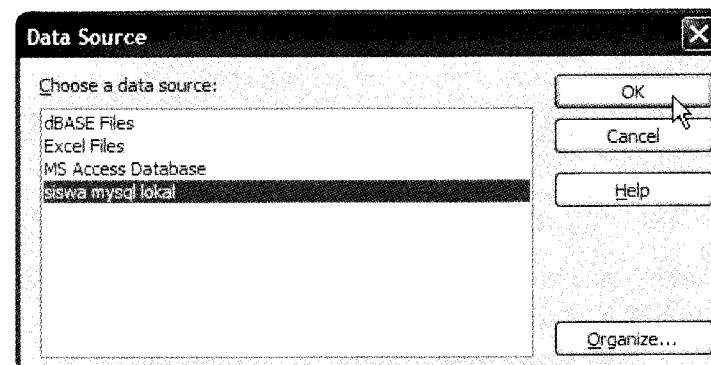
Gambar 6.26 Memilih ODBC sebagai konektor ke MySQL.

5. Di **Setup ODBC Connection**, klik **Browse** untuk memilih data source.



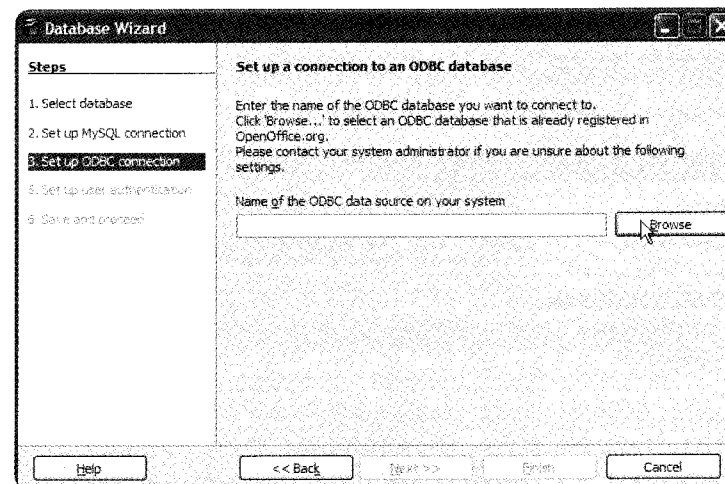
Gambar 6.27 Jendela Setup odbc connection.

6. Pilih sumber data source ke MySQL sesuai dengan yang telah dibuat di user dan sebelumnya. Lalu klik **OK**.



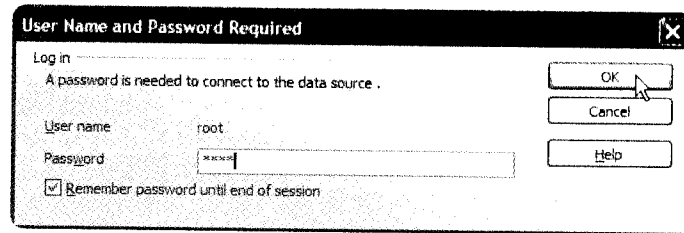
Gambar 6.28 Pemilihan koneksi data source ke MySQL.

7. Isi **Username** dan **Password** untuk menghubungkan ke dsn tersebut. Klik **Test Connection** untuk mengetes koneksi. Jika ada password untuk menghubungkan ke MySQL, cek pada **Password required**.



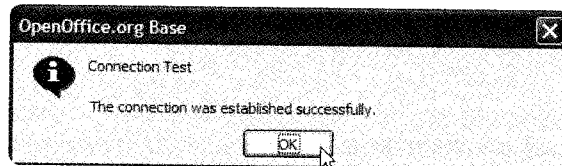
Gambar 6.29 Password required.

8. Isi password pada textbox **Password** lalu klik **OK**.



Gambar 6.30 Pengisian password sebelum tes koneksi.

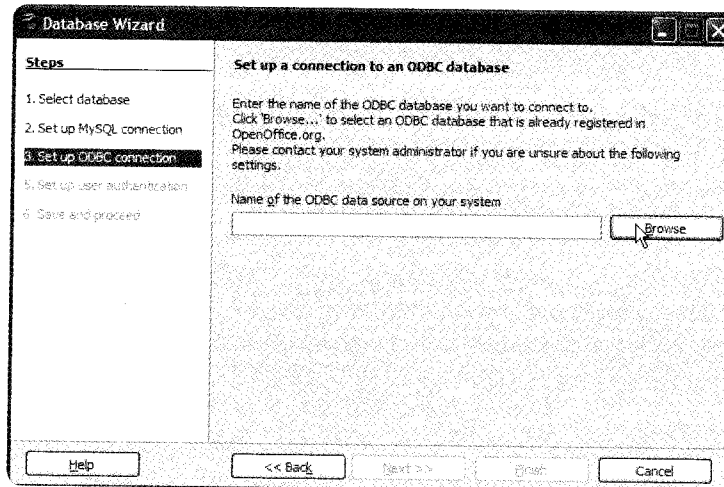
9. Jika koneksi sudah benar, muncul kotak informasi **The connection was established successfully**. Klik **OK**.



Gambar 6.31 Koneksi sudah dites, dan benar

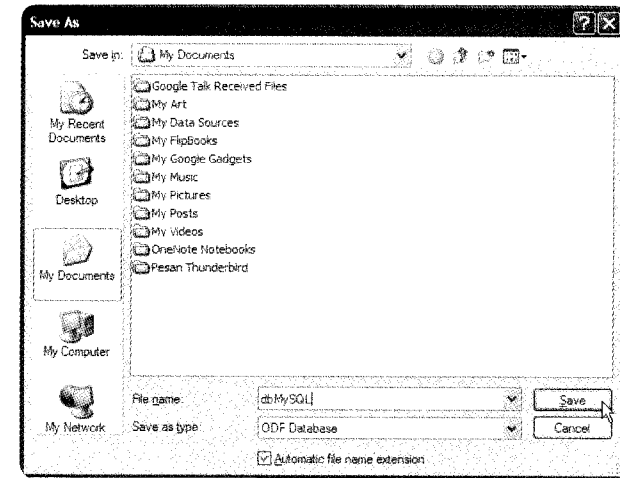
10. Klik **Finish**.

Database yang terbuat akan tersimpan dalam satu file tersendiri, walau pun kontennya nanti merupakan indikasi dari MySQL.



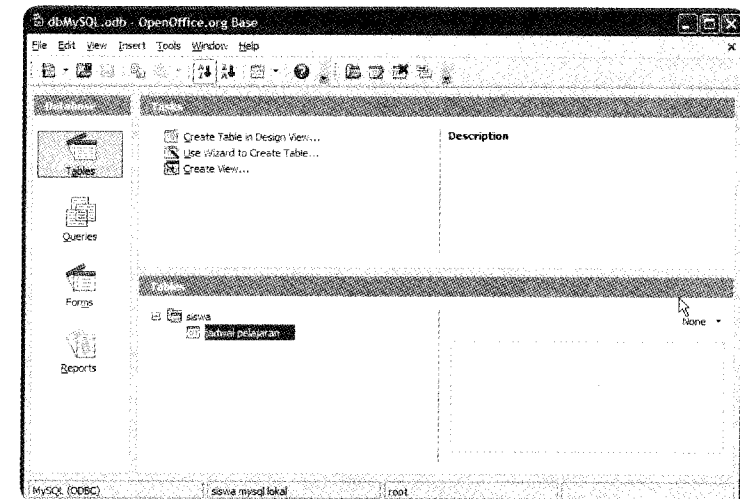
Gambar 6.32 Save and proceed, tahap terakhir dari wizard database di OpenOffice base3.

11. Saat muncul kotak dialog **Save As**, isi nama file database OpenOffice base yang akan Anda simpan lalu klik **Save**.



Gambar 6.33 Penyimpanan database OpenOffice base.

Tampilan OpenOffice base terlihat seperti gambar berikut ini. Di bagian Tables sudah ada tabel baru, itu adalah tabel yang ada di MySQL.

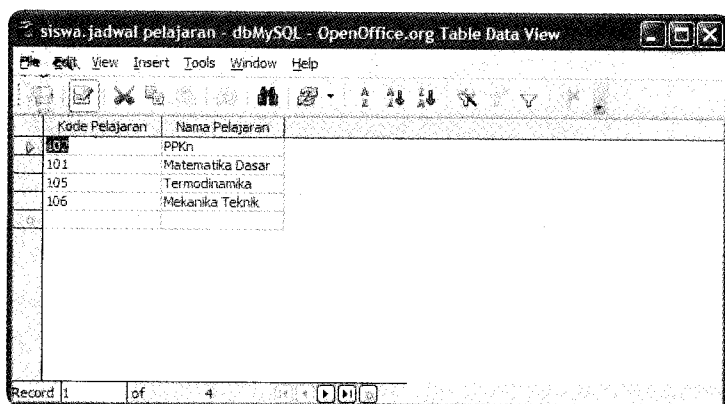


Gambar 6.34 Tabel MySQL.

MENAMBAHKAN DATA KE MYSQL

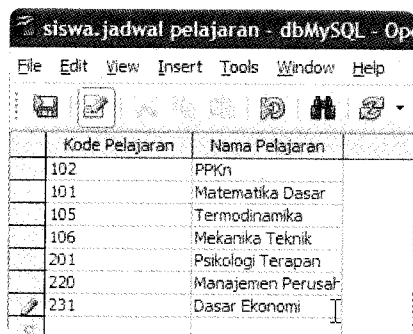
Database di OpenOffice base yang dihubungkan ke MySQL ini bersifat aktif, yaitu OpenOffice base bersifat sebagai interface. Artinya, apa yang dilakukan di OpenOffice base juga akan diterapkan ke database MySQL secara langsung. Anda juga dapat menambahkan data ke MySQL.

1. Klik ganda pada tabel untuk membuka jendela yang berisi konten tabel tersebut.



Gambar 6.35 Tabel terbuka, berisi data-data dari tabel jadwal pelajaran di MySQL.

2. Isi nilai di baris terakhir. Misalnya, ditambah 3 record dengan kode pelajaran 201, 220, dan 231 berikut ini.



Gambar 6.36 Pengisian record baru untuk database MySQL di OpenOffice base.

Ketika perubahan sudah disimpan dengan mengklik ikon bergambar disket di kiri atas toolbar, perubahan langsung diterapkan di MySQL juga. Ini dapat diketahui jika Anda membuka database menggunakan tool lain, seperti PHPMyAdmin.

Urut berdasarkan kunci: tanpa

		Kode Pelajaran	Nama Pelajaran
☐		102	PPKn
☐		101	Matematika Dasar
☐		105	Termodinamika
☐		106	Mekanika Teknik
☐		201	Psikologi Terapan
☐		220	Manajemen Perusahaan
☐		231	Dasar Ekonomi

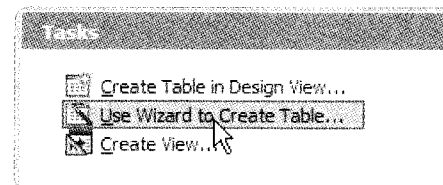
Gambar 6.37 Perubahan database terlihat di PHPMyAdmin.

MEMBUAT TABEL DI MYSQL BARU DENGAN BASE

Salah satu keunggulan Base dibandingkan PHPMyAdmin dalam hal manajemen database adalah kemampuannya untuk membuat tabel dengan bantuan wizard ataupun secara manual. Karena merupakan aplikasi desktop, OpenOffice base memiliki jauh lebih banyak fitur dibandingkan dengan PHPMyAdmin dalam hal manajemen database, walaupun tentu fitur-fitur tertentu seperti dump dan backup ke dalam file plain text tidak ada di Base.

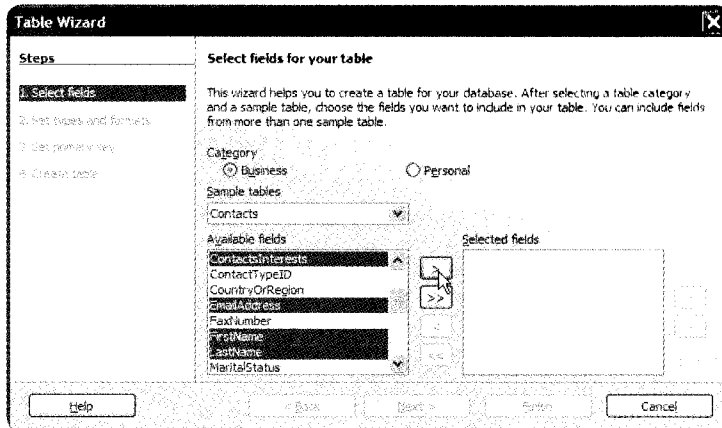
Berikut ini cara membuat tabel baru untuk database MySQL menggunakan fasilitas wizard dari base:

1. Dari jendela utama base, klik link **Use wizard to create table** di bagian kotak **Tasks**.



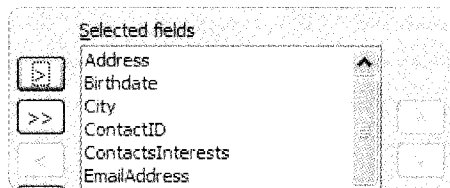
Gambar 6.38 Use wizard to Create Table.

- Pilih field dengan memilih category (Business atau personal) kemudian memilih field-field yang ingin dimasukkan.



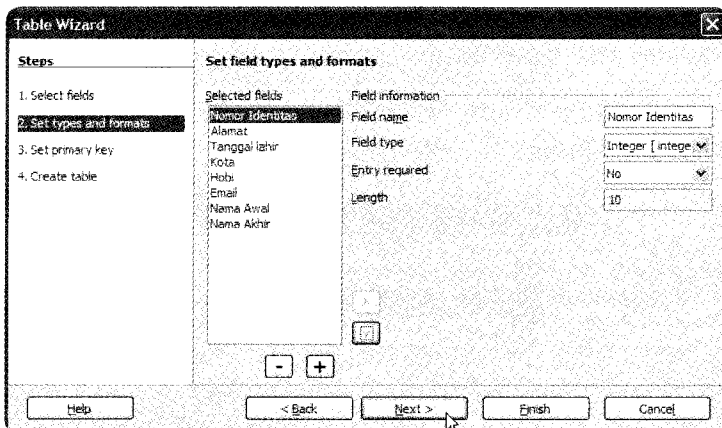
Gambar 6.39 Pemilihan field dari tabel Contacts.

- Klik field lalu klik panah kanan untuk memasukkannya ke kotak **Selected fields**. Klik **Next**.



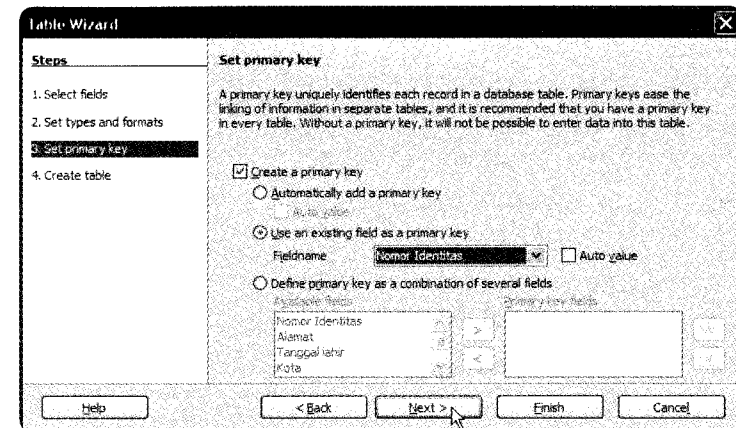
Gambar 6.40 Field-field yang sudah dipilih di kotak Selected fields.

- Ubah nama field atau tipe field, sesuai kebutuhan. Klik **Next**.



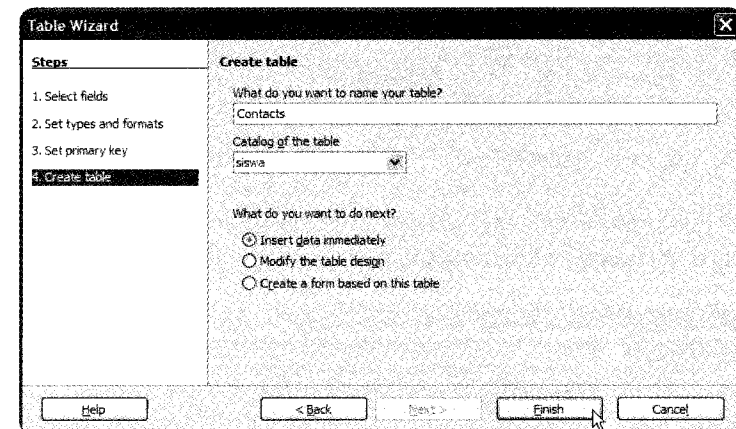
Gambar 6.41 Tampilan field sudah diubah ke dalam bahasa Indonesia.

- Klik **Next**. Langkah ketiga adalah setting primary key, sebagai contoh pilih **Use a existing field as a Primary**.



Gambar 6.42 Penentuan set primary key.

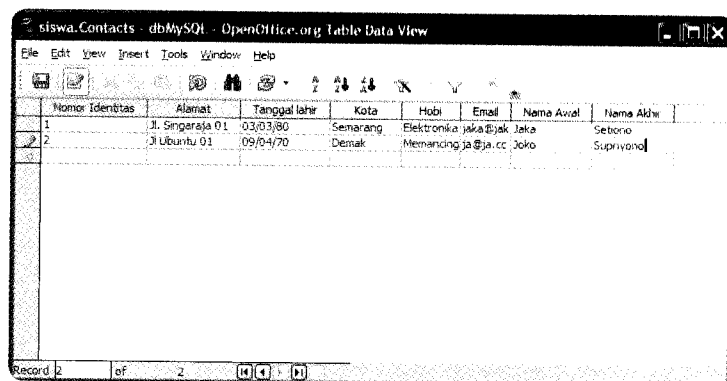
- Pilih **Insert data immediately** lalu klik **Finish**.



Gambar 6.43 Penyimpanan table.

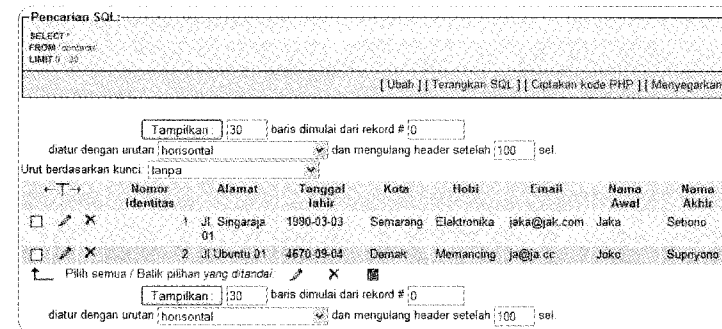
Tabel akan langsung terbuat, kemudian muncul jendela tabel tersebut, yang mengizinkan Anda mengisi data ke dalam database MySQL secara interaktif.

Pengisian dapat dilakukan dengan langsung mengisi ke sel-sel yang ada di tabel tersebut.



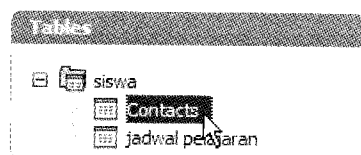
Gambar 6.44 Pengisian tabel MySQL di OpenOffice base.

Jika di browse, tabel tersebut berisi data sesuai dengan yang diisi oleh OpenOffice base. Ini menunjukkan bahwa pembuatan dan pengisian tabel langsung berpengaruh terhadap kondisi database MySQL.



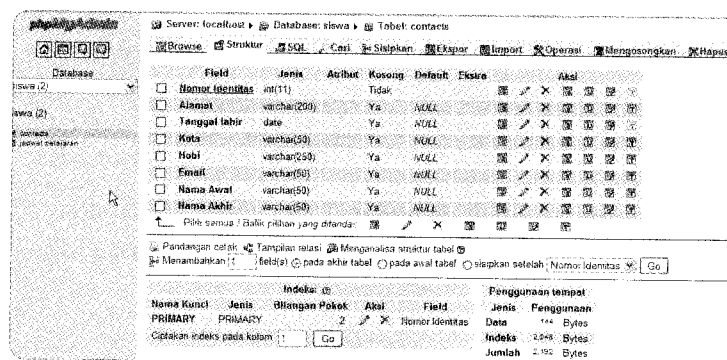
Gambar 6.47 Isi tabel Contacts yang terlihat di database.

Untuk menyimpannya, klik pada ikon Save bergambar disket di jendela Tables. Di kotak tables terlihat ada tabel bernama Contacts.



Gambar 6.45 Tabel Contacts di database.

Ketika dilihat di software lain seperti PHPMyAdmin, juga terlihat adanya tabel baru bernama contacts di tabel Siswa.



Gambar 6.46 Struktur table Contacts yang dibuat oleh OpenOffice base terlihat di PHPMyAdmin.

MANAJEMEN USER

Bab ini akan membahas:

- Penggunaan user management untuk mengatur user di MySQL.
- Pembuatan user dan pengaturan privilege.
- Penggunaan PHPMyAdmin untuk user management.

MySQL 5 merupakan sebuah database multiuser. Untuk itu, user-user perlu diperlakukan dengan berbeda-beda. Dengan demikian, perlu adanya sistem autentifikasi dan manajemen user sehingga tiap user akan memiliki kewenangan yang berbeda-beda.

MENGENAL MODUS AUTENTIFIKASI MYSQL

Untuk memahami bagaimana manajemen user, terlebih dahulu Anda harus tahu bagaimana MySQL melakukan otentifikasi user. Otentifikasi user berfungsi sebagai sistem yang akan mengizinkan atau mencegah user untuk mengakses database tertentu atau tabel tertentu.

Dengan adanya manajemen user yang canggih, Anda dapat membatasi tipe operasi yang bisa dilakukan oleh pengguna terhadap database.

Apa saja yang diperlukan ketika autentifikasi dengan MySQL? Yang diperlukan hanyalah username dan password. Ketika username dan password sudah diisi dengan benar, user tersebut adalah user yang valid dan terautentifikasi sehingga dapat memperoleh akses. Selain itu, ada juga nama host (hostname) yang diperlukan oleh MySQL.

Localhost merupakan nama hostname yang paling umum. Ini menunjukkan bahwa Anda bisa berkoneksi dengan MySQL yang ada di komputer lokal atau MySQL client berada di satu mesin yang sama dengan server MySQL. Bisa juga database diakses dari satu aplikasi atau script yang berjalan di mesin yang sama.

Penentuan hostname bertujuan agar MySQL mengenali lokasi komputer tempat Anda mengakses database. Jika memang lokasi tersebut diizinkan (allowed), barulah koneksi bisa dijalankan.

Bisa seorang dengan username yang sama, tetapi mengakses dari lokasi yang berbeda dengan kewenangan yang berbeda. Jadi, ada 3 variabel yang menentukan seorang user berwenang, yaitu username, password, dan nama host.

Misalnya, ada user bernama Joko. Tidak cukup hanya menyatakan user

tersebut bernama joko, tetapi harus joko@localhost, joko@wahana-kom.com dan seterusnya.

Ada beberapa privilege dari MySQL yang menentukan apa saja yang mungkin dilakukan oleh user. Misalnya, user yang punya privilege SELECT dapat melakukan query pada tabel. Adapun jika memiliki privilege INSERT, barulah bisa memasukkan data.

Untuk mengatur privilege, user perlu diberikan (granted). Untuk itu, nanti akan dijelaskan penggunaan perintah GRANT dan REVOKE. Berikut ini beberapa privilege yang bisa dialokasikan ke user.

Tabel 7.1 Daftar privilege user di MySQL

ALTER	Memungkinkan ALTER TABLE.
ALTER ROUTINE	Memungkinkan perubahan stored routine.
CREATE	Bisa melakukan CREATE TABLE.
CREATE ROUTINE	Bisa membuat stored routine.
CREATE TEMPORARY	Bisa melakukan CREATE TEMPO TABLE.
CREATE USER	Bisa melakukan CREATE USER, DROP USER, RENAME USER, dan REVOKE ALL PRIVILEGES.
CREATE VIEW	Bisa melakukan CREATE VIEW.
DELETE	Bisa melakukan DELETE.
DROP	Bisa melakukan DROP TABLE.
EXECUTE	Bisa mengeksekusi stored routine.
FILE	Bisa menggunakan SELECT... INTO OUTFILE dan LOAD DATA INFILE.
INDEX	Bisa menggunakan CREATE INDEX dan DROP INDEX.

INSERT	Bisa menggunakan INSERT.
LOCK TABLES	Bisa memakai LOCK TABLES di tabel di mana user memiliki privilege SELECT.
PROCESS	Bisa menggunakan SHOW FULL PROCESSLIST.
RELOAD	Bisa menggunakan FLUSH.
REPLICATION	Bisa memungkinkan user untuk menanyakan di mana slave atau master
CLIENT	Kondisi server
REPLICATION SLAVE	Diperlukan untuk replikasi slave.
SELECT	Memungkinkan menggunakan SELECT.
SHOW DATABASES	Memungkinkan penggunaan SHOW DATABASES.
SHOW VIEW	Memungkinkan penggunaan SHOW CREATE VIEW.
SHUTDOWN	Memungkinkan penggunaan mysqladmin shutdown.
SUPER	Memungkinkan penggunaan CHANGE MASTER, KILL, PURGE MASTER LOGS, dan SET GLOBAL SQL statment. Memungkinkan debug mysqladmin. Memungkinkan satu koneksi tambahan jika koneksi maksimum sudah terpenuhi.
UPDATE	Memungkinkan adanya UPDATE.
USAGE	Memungkinkan adanya koneksi tanpa privilege khusus.

Di manakah privilege MySQL disimpan? Jawabannya adalah di tabel privilege MySQL. Setiap kali akses ke MySQL diberikan ke user, ada perubahan di database.

Akses ke database MySQL tersebut hanya bisa dilakukan oleh superuser, yaitu yang memiliki akses root saja. Hanya superuser yang bisa membuka tabel privilege. Data dari tabel user ini digunakan untuk otentifikasi. Kolom host, user, dan password mengandung nilai yang digunakan untuk validasi koneksi. Namun, kolom password dienkripsi sehingga Anda tidak bisa mengintip password user di sini.

		Host	User	Password	
☐		☒	localhost	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
☐		☒	127.0.0.1	root	
☐		☒	localhost	db_user	*F8F29CEDC6CF25F8B020A118D7A150E460661166

Gambar 7.1 Contoh isi tabel MySQL.user.

Tabel db berisi semua permission asli yang dimiliki oleh tiap user terhadap database. Isinya ada kolom db, user, host kemudian ada juga kolom-kolom yang menentukan privilege lainnya seperti Select_priv dan Alter_priv. Nilai-nilai kolom ini bisa Y (Yes) atau N (No). Nilai Y atau N mengindikasikan apakah privilege-privilege tersebut sudah diberikan (granted) atau belum.

Sementara privilege untuk masing-masing tabel disimpan di tabel bernama tables_priv. Tabel ini menggunakan kolom Db, User, Host, dan Table_name untuk mengidentifikasi tabel dari database dan user. Kolom Table_priv berisi nama dari jenis privilege yang diberikan, sedangkan kolom db berisi satu baru per user dan database. Ada satu baris di tables_priv untuk tiap kewenangan setingkat tabel yang diberikan.

Adapun cara mengubah privilege sebenarnya bisa dengan langsung mengubah nilai-nilai di tabel yang bersangkutan di atas, tetapi tentu tidak efisien dan lebih sulit. Karena itu, yang umum digunakan adalah menggunakan command akan dijelaskan berikutnya di bab ini. Alasan lain kenapa tidak disarankan mengubah langsung di tabel adalah karena perubahan terhadap tabel privilege tidak akan diterapkan kecuali setelah Anda mengakses command Flush privileges.

MANAJEMEN USER

Manajemen user adalah tentang bagaimana membuat dan mengatur user-user yang ada di MySQL. Teknisnya adalah bagaimana manajemen account dan pemberian (grant) dan mencabut (revoke) privilege.

PEMBUATAN USER

Perintah yang digunakan untuk membuat user baru adalah CREATE USER. Parameter minimal yang harus diberikan adalah nama user itu sendiri alias username. Nama user yang diberikan harus berupa nama user dan nama host dalam format nama_user@host. Kemudian diikuti dengan command IDENTIFIED BY yang dilanjutkan dengan password untuk user tersebut. Password merupakan kata sandi yang digunakan bagi user untuk login. Seandainya Anda tidak mengalokasikan password untuk user, user tersebut akan selalu bisa login ke MySQL tanpa harus mengisi password sama sekali. Ini tentu tidak aman, kecuali jika memang ingin yang praktis digunakan di lingkungan yang aman, seperti di LAN atau di komputer lokal.

Berikut ini contoh perintah untuk membuat user bernama 'joko' yang bisa mengakses database MySQL dari localhost dengan menggunakan password 'mas_joko':

```
mysql> CREATE USER joko@localhost IDENTIFIED BY
'mas_joko';
```

```
Query OK, 0 rows affected (0.01 sec)
```

User yang bisa membuat user dalam kondisi awal adalah superuser alias root. Namun, Anda juga bisa memberi izin ke user lain (non-root) untuk bisa membuat user baru dengan memberikan user tersebut privilege agar bisa menjalankan command CREATE USER.

Selain itu, Anda juga harus memberikan user tersebut kewenangan untuk bisa melakukan INSERT di database MySQL. Karena operasi dasar untuk membuat user adalah menggunakan insert command ke record di tabel mysql.user.

SETTING PASSWORD

Nilai password disimpan di kolom password di tabel mysql.user, tetapi penyimpanannya tidak berupa teks yang bisa terbaca, melainkan dienkripsi menggunakan algoritma hash sehingga tidak bisa dibalikkan lagi ke password semula sehingga aman. Anda bisa melakukan hash yang sama menggunakan fungsi PASSWORD().

Untuk mengatur password ke user, gunakan perintah SET PASSWORD. Misalnya, untuk memberi password baru ke user joko@localhost, Anda bisa menggunakan contoh kode berikut:

```
mysql> SET PASSWORD FOR joko@localhost =
PASSWORD('password_baru');
```

```
Query OK, 0 rows affected (0.01 sec)
```

Jika command PASSWORD() dilupakan, MySQL akan memperingatkan bahwa Anda tidak memasukkan password yang valid. Misalnya, seperti berikut ini:

```
mysql> SET PASSWORD FOR joko@localhost =
'password_baru';
```

```
ERROR 1372 (HY000): Password hash should be a
41-digit hexadecimal number
```

Siapakah yang bisa mengganti password seorang user? Jawabannya adalah user itu sendiri dan root. Setiap user bisa mengubah password-nya sendiri, tetapi tidak bisa mengubah password user lain. Hanya root yang bisa mengubah semua password yang ada di MySQL.

Anda juga bisa mengganti password langsung ke user yang sedang login ke MySQL dengan menghilangkan kata FOR menjadi seperti berikut:

```
mysql> SET PASSWORD = PASSWORD('password_
baru');
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
Granting Privileges
```

GRANT PRIVILEGE

Untuk memberikan kewenangan tertentu bagi user, Anda bisa menggunakan command GRANT. Command GRANT ini bisa diberikan kepada satu atau banyak kewenangan, atau bisa juga untuk semua kewenangan (ALL PRIVILEGES).

Berikut ini contoh penggunaan perintah GRANT kepada user joko@localhost untuk mengalokasikan kewenangan SELECT dan INSERT untuk tabel 'contacts' di database 'siswa'.

```
mysql> GRANT SELECT, INSERT
      -> ON siswa.contacts
      -> TO joko@localhost;

Query OK, 0 rows affected (0.01 sec)
```

User yang tidak memiliki kewenangan Select tidak bisa ditampilkan jika perintah show tables dieksekusi. Jika sebuah user bisa mengeksekusi SELECT dan INSERT, user tersebut bisa membuka tabel dan memasukkan nilai. Namun, ketika hendak menggunakan command yang kewenangannya tidak dimiliki, akan ada pemberitahuan bahwa tidak ada permission alias **permission denied**.

```
mysql> INSERT INTO 'jadwal pelajaran'
      ->VALUES (303, 'Mekanika Teknik');

Query OK, 1 row affected (0.01 sec)
```

Ketika menggunakan perintah Delete, contohnya seperti berikut:

```
mysql> DELETE FROM contacts;

ERROR 1142 (42000): DELETE command denied to
user
'joko'@'localhost' for table 'contacts'
```

Sebuah privilege yang dimiliki user akan bisa membuat user tersebut mengutak-atik database sesuai dengan privilege yang dimilikinya.

Namun, user tersebut tetap tidak bisa memberikan privilege-nya kepada user lain kecuali memang diberikan hak untuk itu.

Untuk bisa mengakomodasi hal ini, Anda bisa menggunakan command WITH GRANT OPTION di statement GRANT sebelumnya. Contoh berikut yang memungkinkan joko@localhost untuk bisa memberikan grant SELECT untuk user lain.

```
mysql> GRANT SELECT
      -> ON siswa.contacts
      -> TO joko@localhost
      -> WITH GRANT OPTION;

Query OK, 0 rows affected (0.01 sec)
```

Walaupun perintah SELECT nya sama seperti sebelumnya, di akhir statement ada statement WITH GRANT OPTION. Ini berarti user joko@localhost bisa memberikan privilege ini ke orang lain yang diinginkannya.

MENCABUT PRIVILEGE

Kebalikan dari GRANT, Anda juga bisa mencabut privilege dari user. Cara untuk mencabut privilege adalah menggunakan perintah REVOKE. Sintaks dari REVOKE sendiri mirip dengan GRANT. Misalnya, perintah berikut untuk menghilangkan kewenangan INSERT yang sudah diberikan di contoh sebelumnya:

```
mysql> REVOKE INSERT
      -> ON siswa.contacts
      -> FROM joko@localhost;

Query OK, 0 rows affected (0.25 sec)
```

Jika hanya satu privilege yang dicabut, privilege lain di user yang bersangkutan tidak akan berubah. Misalnya, untuk user joko@localhost di atas, privilege SELECT nya tidak akan hilang.

Anda juga bisa memberikan privilege ke username dengan menggunakan wildcard, menggunakan karakter % atau _ (underscore). Contohnya begini, Anda ingin username dengan password tertentu bisa mengakses database tanpa memperhatikan host-nya, apakah localhost atau tidak. Untuk itu, bisa menggunakan karakter wildcard.

Contohnya kode berikut:

```
mysql> CREATE USER 'joko@%' IDENTIFIED BY
'password_baru';
```

```
Query OK, 0 rows affected (0.01 sec)
```

Kode di atas akan membuat user joko di semua hostname dengan password = password_baru. Jadi, user joko ini bisa mengakses database dari semua host menggunakan password 'password_baru'.

```
mysql> GRANT SELECT, INSERT
-> ON siswa.contacts
-> TO 'joko@%';
```

```
Query OK, 0 rows affected (0.01 sec)
```

Jangan lupa, nama user yang ingin diberi wildcard harus dibuka dan tutup oleh tanda petik tunggal. Jika tidak, tidak akan dianggap sebagai sebuah wildcard dan akan menjadi error.

Karakter wildcard lain adalah underscore yang contoh penerapannya adalah untuk menuliskan IP address dalam rentang tertentu. Misalnya:

```
mysql> CREATE USER 'joko@192.168.0._'
-> IDENTIFIED BY 'password_baru';
```

```
Query OK, 0 rows affected (0.01 sec)
```

Dalam contoh di atas, user akan bisa terautentifikasi sebagai joko dengan password password_baru jika menghubungkan ke database dari IP Address antara 192.168.0.1 sampai 192.168.0.9.

MELIHAT PRIVILEGE USER

Untuk melihat kewenangan-kewenangan apa saja yang dimiliki user, Anda bisa menggunakan perintah SHOW GRANTS. Hasilnya adalah daftar statement grant yang bisa digunakan untuk membuat user dengan privilege yang sama.

Contoh penggunaannya adalah:

```
mysql> SHOW GRANTS FOR joko@localhost;
```

MENGHAPUS USER

Untuk menghapus user, Anda bisa menggunakan perintah DROP USER yang contohnya seperti berikut:

```
mysql> DROP USER joko@localhost;
```

```
Query OK, 0 rows affected (0.00 sec)
```

Di versi MySQL sebelum 5.0, Anda harus me-revoke semua privilege yang ada di user baru, kemudian menghapusnya secara manual dari tabel MySQL.user/

Jadi ini contoh di versi MySQL lama:

```
mysql> REVOKE ALL PRIVILEGES
```

```
-> ON *.*
```

```
-> FROM joko@localhost;
```

```
Query OK, 0 rows affected (0.01 sec)
```

Setelah semua privilege dicabut, barulah isi di database MySQL.user dihapus menggunakan perintah:

```
mysql> DELETE FROM mysql.user
```

```
-> WHERE User = 'joko'
```

```
-> AND Host = 'localhost';
```

```
Query OK, 1 row affected (0.00 sec)
```

Setelah itu, agar privilege ter update, gunakan flush untuk menerapkan perubahan tersebut.

```
mysql> FLUSH PRIVILEGES;
```

Query OK, 0 rows affected (0.01 sec)

Agar bisa menggunakan perintah DROP USER, Anda harus memiliki privilege CREATE USER. Wajar saja, karena yang bisa menghapus harus bisa menciptakan juga. Namun, jika bukan superuser/root, Anda harus memiliki kewenangan DELETE di tabel MySQL.user.

MANAJEMEN USER DI PHPMYAdmin

PHPMyAdmin memiliki fitur yang sangat memudahkan manajemen user. Anda dapat membuat sekaligus mengalokasikan privilege secara lebih praktis dan cepat.

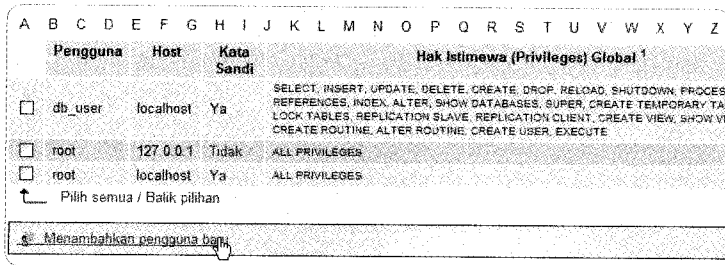
Berikut ini cara membuat user baru di PHPMyAdmin beserta cara untuk mengolah hak akses/privilege-nya:

1. Login ke PHPMyAdmin sebagai superuser atau user lain yang bisa melakukan pembuatan user baru. Lalu klik **Hak Akses** di dasbor PHP-MyAdmin.



Gambar 7.2 Tab Hak akses di PHPMyAdmin 3.

2. Klik link **Menambahkan pengguna baru**.



Gambar 7.3 Link untuk menambahkan pengguna baru.

3. Isi **Username**, kemudian **Host**, lalu kata sandi.

Menambahkan pengguna baru

Gambar 7.4 Pengisian informasi user baru

4. Pilih **Create database with same name and grant all privileges**.

Gambar 7.5 Pemilihan database

5. Klik **Pilih semua** untuk memudahkan.

Gambar 7.6 Pemilihan privilege.

6. Klik **Go**.

User, database, sekaligus privilege sudah terbuat. Mudah, bukan? Sangat nyaman dibandingkan dengan mengetikkan menggunakan sql manual. Otomatis ada sql statement yang di-generate oleh PHPMyAdmin.

```
CREATE USER 'joko'@'localhost' IDENTIFIED BY
'****';

GRANT ALL PRIVILEGES ON * . * TO
'joko'@'localhost' IDENTIFIED BY '****' WITH
GRANT OPTION MAX_QUERIES_PER_HOUR 0 MAX_
CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0 ;

CREATE DATABASE IF NOT EXISTS `joko` ;

GRANT ALL PRIVILEGES ON `joko` . * TO
'joko'@'localhost';
```

Ketika dibuka kembali, terlihat ada user baru di tabel Pengguna.

Pengguna	Host	Kata Sandi	Hak Istimewa (Privileges) Global ¹
db_user	localhost	Ya	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EXECUTE
joko	localhost	Ya	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EXECUTE
root	127.0.0.1	Tidak	ALL PRIVILEGES
root	localhost	Ya	ALL PRIVILEGES

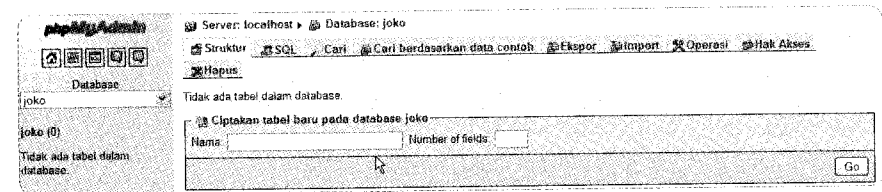
Gambar 7.7 Tabel pengguna terlihat ada user baru sesuai dengan username yang dibuat.

Apabila isi tabel MySQL.db dibuka, terlihat isi halaman berikut ini:

Options									
	Host	Db	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	
☐	localhost	db_user_%	db_user	Y	Y	Y	Y	Y	
☐	localhost	joko	joko	Y	Y	Y	Y	Y	

Gambar 7.8 Halaman terbuka.

Ada pula database yang terbuat sesuai dengan nama user yang dibuat. Ini adalah fitur khusus dari PHPMyAdmin yang mengotomasi semuanya dengan mudah dan praktis.



Gambar 7.9 Database baru bernama joko.

OPTIMASI MySQL

Bab ini akan membahas:

- Penyetingan platform optimal.
- Pengenalan konfigurasi hardware optimal MySQL.
- Mengenal optimasi struktur tabel.
- Mengenal penggunaan perintah flush.

Dengan melakukan pengaturan MySQL pada posisi optimalnya, Anda akan mendapatkan layanan MySQL yang sempurna dan tanpa kegagalan. Untuk bisa mengoptimalkan MySQL, Anda perlu tahu beberapa hal, dari mulai konfigurasi hardware hingga perintah-perintah tertentu untuk mengetes kondisi MySQL.

KONFIGURASI HARDWARE

Mendesain database yang berstruktur baik dan normal, barulah separuh dari yang bisa dilakukan untuk mengoptimalkan MySQL. Agar MySQL bisa berjalan kencang dan optimal setidaknya diberikan hardware dengan spesifikasi yang cukup.

Untuk mengoptimalkan MySQL, paling tidak ada 4 komponen komputer yang perlu diperhatikan. Yaitu prosesor (CPU), harddisk, memory, dan sistem operasi. Pemilihan empat komponen yang tepat ini bisa membuat MySQL berjalan dengan lebih optimal sehingga database bisa lebih cepat aksesnya.

- CPU: Semakin cepat CPU, MySQL akan bisa memproses data lebih cepat. Cara memilih prosesor sebenarnya mudah, tinggal cari yang frekuensinya paling tinggi. Dengan harga prosesor yang makin murah, kecepatan di atas 1GHz sekarang sudah bisa diperoleh dengan sangat terjangkau. Karena itu, carilah prosesor yang cukup besar untuk menjalankan MySQL Anda.
- Memory: Semakin banyak kapasitas RAM di komputer, akan mempercepat MySQL. Harga RAM sekarang (2009) sangat jatuh sehingga menggunakan RAM hingga 2G atau lebih tak masalah. RAM yang besar juga cocok digunakan jika komputer Anda menggunakan CPU/prosesor yang lambat, karena RAM bisa mengimbangi kinerja prosesor yang lambat.
- Harddisk: Pilihlah harddisk yang cukup besar kapasitasnya tapi juga cepat untuk mengakomodasi kebutuhan MySQL dan lalu lintas datanya. Parameter yang digunakan untuk menilai harddisk adalah waktu pencariannya (seek time) atau waktu yang diperlukan bagi

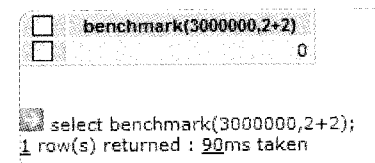
harddisk untuk berputar dan mencari data. Waktu pencarian data ini sekitar delapan atau sembilan milidetik. Jangan lupa pula kapasitas harus besar sehingga bisa mengakomodasi semua penyimpanan data tapi juga masih cepat untuk mencari datanya.

- Sistem operasi: Sistem operasi yang digunakan bisa berpengaruh terhadap penggunaan RAM dan prosesor. Untuk optimalnya adalah menggunakan sistem operasi yang tidak boros RAM. Misalnya, sistem operasi yang berbasis command line saja dan tidak memiliki antarmuka (GUI), seperti Linux tapi tanpa GNOME atau KDE. Karena itulah biasanya untuk traffic yang super padat, orang lebih suka menggunakan Linux daripada Windows. Namun, itu semua tidak signifikan jika memang konfigurasi hardware sudah sangat tinggi sehingga Windows ataupun GUI dari sistem operasi tidak mengonsumsi sumber daya hardware secara berlebihan.

Untuk melihat kecepatan pemrosesan MySQL, Anda bisa menggunakan perintah benchmark. Perintah ini akan melihat berapa cepat MySQL memproses query benchmark. Benchmark ini dimasukkan setelah statement select tapi nantinya tidak akan menghasilkan nilai apapun, karena tujuannya hanya berapa waktu untuk melakukan ekspresi berulang dalam rentang waktu tertentu. Berikut ini contoh benchmark untuk mengecek perintah penjumlahan 2+2 selama tiga juta kali.

```
mysql> select benchmark(3000000,2+2);
```

Hasilnya bisa dilihat di gambar berikut:



Gambar 8.1 Hasil benchmark untuk menandai kecepatan pemrosesan MySQL.

Semakin cepat waktu pemrosesan, akan semakin optimal pula MySQL-nya. Sering kali kecepatan benchmark ini berbeda-beda tergantung kepada traffic dan jumlah pengguna yang sedang mengakses database.

OPTIMASI STRUKTUR TABEL

Cara lain mempercepat akses adalah dengan melakukan optimasi struktur tabel. Optimasi struktur tabel berbeda dengan optimasi melalui pembuatan tabel dengan desain yang baik. Yang dimaksud optimasi struktur tabel di sini adalah optimasi dalam penggunaan ruang kosong yang tak lagi dipakai setelah penghapusan data di tabel. Lebih tepatnya, bersihkan tabel setelah ada utak-atik struktur tabel.

Untuk melakukan optimasi struktur tabel, digunakan perintah OPTIMIZE TABLE diikuti dengan nama tabel. Misalnya, ketika ingin mengoptimalkan tabel contacts di database siswa, Anda dapat menggunakan perintah.

```
mysql> optimize table contacts;
```

☐	Table	Op	Msg_type	Msg_text
☐	siswa.contacts	optimize status		Table is already up to date

Gambar 8.2 Hasil optimasi

Ketika menjalankan perintah tersebut, hasil di kolom Msg_text hanya Table is **already up to date**.

PERINTAH EXPLAIN SELECT

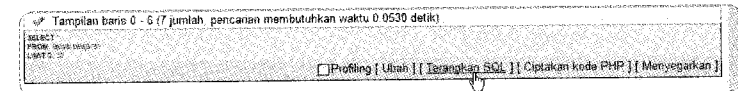
Keluaran dari perintah EXPLAIN adalah sebuah tabel yang mengandung kolom-kolom berikut:

- Table: nama tabel.
- Type: tipe join.
- possible_keys: kolom ini menunjukkan indeks MySQL yang digunakan untuk mencari baris tertentu di tabel. Jika hasilnya null, tidak ada indeks yang bisa dipakai untuk membantu query. Adanya indeks bisa mempercepat performa dari sebuah query.
- Key: key yang digunakan di query, nilainya juga Null jika tak ada indeks yang dipakai.
- key_len: panjang key yang dipakai, jika ada.

- Ref: kolom yang digunakan dengan key untuk mengambil hasilnya.
- Rows: jumlah row dari MySQL yang harus diperhatikan untuk mengeksekusi query.
- Extra: informasi tambahan mengenai bagaimana MySQL akan mengeksekusi query. Ada beberapa opsi, seperti menggunakan index dan WHERE.

```
mysql> explain select * from 'jadwal pelajaran'
```

Atau, jika di PHPMyAdmin, Anda bisa mengklik link Terangkan SQL setelah mem-browse sebuah data atau mengeksekusi query tertentu.



Gambar 8.3 Link Terangkan SQL untuk mengeksekusi command Explain.

Contoh hasil explain menjadi seperti berikut:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	jadwal pelajaran	ALL	NULL	NULL	NULL	NULL	7	

Gambar 8.4 Hasil Explain.

Tabel explain tidak menghasilkan sesuatu yang istimewa karena bentuk sql statement yang dijelaskan hanya sederhana.

```
EXPLAIN SELECT *  
  
FROM 'jadwal pelajaran'  
  
WHERE 'Kode Pelajaran' =101
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	jadwal pelajaran	const	PRIMARY	PRIMARY	4	const	1	

Gambar 8.5 Hasil explain ketika ada statement WHERE.

Bagaimana explain digunakan untuk optimasi? Anda tinggal melihat nilai di kolom **rows**. Kolom ini merupakan jumlah baris yang dievaluasi untuk bisa memperoleh nilai. Semakin sedikit jumlah nilai di kolom ini menunjukkan kualitas SQL Statement semakin oke.

Jadi, dengan memilih sql statement yang hanya perlu mengevaluasi sedikit baris, proses pengambilan data lebih cepat disamping akan meringankan kinerja server MySQL karena tidak perlu banyak bekerja untuk mengambil data.

MENGGUNAKAN FLUSH

User dengan kewenangan reload untuk database tertentu bisa menggunakan perintah FLUSH untuk membersihkan cache internal yang digunakan oleh MySQL. Namun, umumnya yang bisa menggunakan flush hanyalah superuser alias root.

Sintaks dari perintah flush ini adalah:

```
FLUSH opsi_flush
```

Opsi umum yang digunakan oleh perintah flush ini antara lain:

- PRIVILEGES
- TABLES
- HOSTS
- LOGS

Dengan mengeksekusi perintah flush secara periodik tiap kali melakukan pengolahan terhadap privilege, table, host atau log, Anda seolah 'menyegarkan kembali' MySQL Anda dengan cara melakukan bersih-bersih pada media penyimpanan internal MySQL.

Perintah FLUSH TABLES akan menutup semua tabel yang sedang dibuka selama beberapa milidetik dan kemudian membukanya lagi. Dengan demikian, MySQL bisa rehat sejenak. Ketika cache penyimpanan ini sudah kosong, MySQL akan bisa menggunakan memory dengan lebih baik sehingga respon terhadap query yang akan dimasukkan menjadi lebih baik.

Perintah FLUSH HOSTS hanya berfungsi mengosongkan cache host. Cache host yang penuh bisa membuat user yang hendak menghubungkan ke database menjadi terputus koneksinya. Untuk itu, reset host dengan menggunakan perintah FLUSH HOSTS ini.

Perintah FLUSH LOGS berguna untuk menutup kemudian membuka kembali semua file log. Jika semua file menjadi beban, Anda bisa menggunakan perintah ini untuk membuat file baru yang kosong dari catatan log. Untuk server yang berat, mungkin seminggu sekali atau sebulan sekali harus dilakukan FLUSH LOGS ini.

MENGGUNAKAN SHOW

Teknik lain untuk mengetahui kondisi server MySQL adalah menggunakan perintah SHOW. Perintah ini banyak opsinya sehingga bisa melakukan banyak hal. Dari mulai memasukkan informasi tentang database MySQL, user, tabel, ataupun privilege. Contoh penggunaan perintah show antara lain:

```
SHOW GRANTS FOR nama_user
```

```
SHOW DATABASES
```

```
SHOW [OPEN] TABLES [FROM nama_database]
```

```
SHOW CREATE TABLE nama_tabel
```

```
SHOW [FULL] COLUMNS FROM nama_tabel [FROM nama_database]
```

```
SHOW INDEX FROM nama_tabel
```

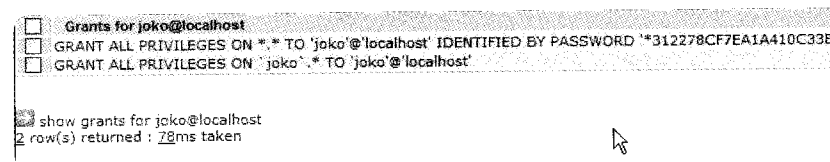
```
SHOW TABLE STATUS
```

```
SHOW STATUS
```

```
SHOW VARIABLES
```

Beberapa perintah SHOW hanya bisa diakses oleh superuser yang memiliki akses ke semua database.

Perintah SHOW GRANTS misalnya, akan menampilkan kewenangan yang dimiliki user tertentu pada host tertentu pula. Ini cocok untuk melihat status dari user tertentu. Biasanya perintah ini dieksekusi sebelum admin/root hendak mengubah privilege user tertentu.

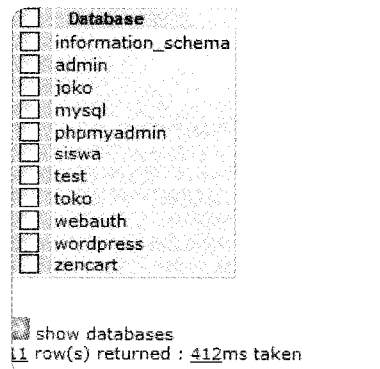


Gambar 8.6 Contoh SHOW grants for user tertentu.

Yang kedua adalah SHOW DATABASES yang fungsinya untuk menampilkan semua daftar database yang ada di MySQL server.

Contohnya seperti berikut:

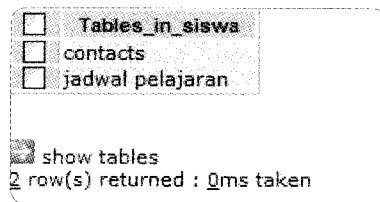
```
mysql> show databases;
```



Gambar 8.7 Contoh perintah Show Databases.

Adapun show tables digunakan untuk melihat tabel-tabel yang ada di database tertentu.

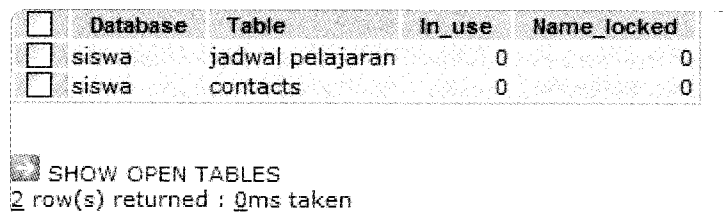
```
mysql> show tables;
```



Gambar 8.8 Contoh perintah show tables.

Jika Anda menambahkan command OPEN di SHOW TABLES, Anda juga bisa melihat semua tabel yang terbuka, artinya yang ada di cache tabel.

```
mysql> SHOW OPEN TABLES;
```

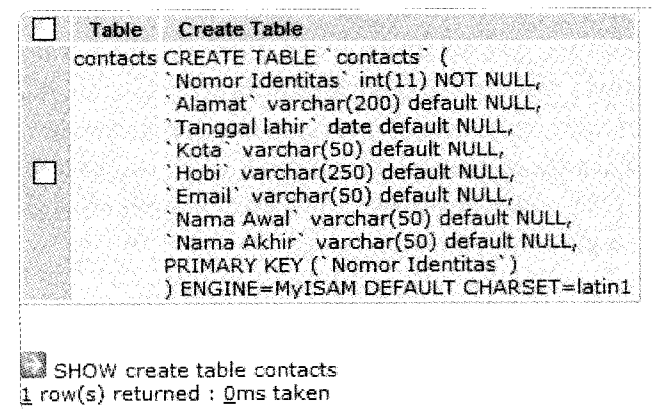


Gambar 8.6 Contoh SHOW grants for user tertentu.

Misalnya, suatu ketika Anda menjumpai jumlah tabel yang dibuka banyak sekali, Anda bisa me-refresh dengan menggunakan FLUSH TABLES sehingga kinerja MySQL tidak terbebani dengan berat. Perintah FLUSH TABLES akan mengosongkan memory sehingga MySQL bisa sedikit bernafas lega.

Anda juga bisa mencari informasi menggunakan SHOW CREATE TABLE, ini akan menunjukkan sql statement yang digunakan untuk membuat tabel tertentu.

```
mysql> show create table contacts;
```



Gambar 8.10 Show create table untuk siswa.contacts.

Dengan informasi yang didapat di sini, Anda bisa mengcopy-paste kode SQL yang dihasilkan untuk membuat tabel yang semisal dengan itu. Adapun untuk mengetahui kolom-kolom yang ada di tabel, Anda bisa menggunakan perintah SHOW COLUMNS FROM seperti berikut:

```
mysql> show columns from contacts;
```

Field	Type	Null	Key	Default	Extra
Nomor Identitas	int(11)	NO	PRI		
Alamat	varchar(200)	YES			
Tanggal lahir	date	YES			
Kota	varchar(50)	YES			
Hobi	varchar(250)	YES			
Email	varchar(50)	YES			
Nama Awal	varchar(50)	YES			
Nama Akhir	varchar(50)	YES			

show columns from contacts
8 row(s) returned : 2ms taken

Gambar 8.11 Perintah Show Columns.

Penutup

Setelah selesai membaca buku ini, penulis yakin Anda sudah menguasai sebagian besar penguasaan MySQL dari mulai instalasi, setting, manajemen user, dan pengolahan database dan tabel.

Walaupun tidak semua detil skenario bisa dijelaskan (karena jika dijelaskan semua malah niscaya buku ini akan sangat tebal), tetapi penulis yakin Anda dapat sedikit berimprovisasi berangkat dari kasus-kasus yang diajarkan di buku ini. Terapkanlah contoh studi kasus yang ada di buku ini ke dalam apa yang Anda hadapi di kantor dengan menggunakan kreativitas dan improvisasi.

Penulis berharap Anda dapat mempergunakan MySQL untuk menunjang kegiatan Anda dengan cara mempermudah Anda mencatat dan mengatur catatan Anda. Apakah Anda karyawan perkantoran, bagian EDP, hobbies, mahasiswa, atau programmer profesional, semoga buku ini membawa manfaat bagi Anda.

Jika pembaca mengalami kendala, pembaca bisa melihat situs resmi MySQL yang berisi contoh-contoh penggunaan dan implementasi MySQL dan studi kasus di mana MySQL sangat berperan penting dalam kegiatan sehari-hari, dari mulai kegiatan akademis hingga bisnis.

Selain itu, Anda juga bisa melihat situs-situs programming yang membahas penggunaan MySQL ataupun pemrograman secara umum baik dari dalam negeri maupun luar negeri. Misalnya, dari ilmukomputer.com. Jika pembaca mengalami kesulitan dalam pembahasan buku ini, pembaca bisa berkonsultasi langsung dengan kami melalui e-mail di litbang@wahanakom.com.

Tentang Penulis

Wahana Komputer-Semarang merupakan suatu perusahaan yang bergerak di bidang teknologi informasi. Wahana Komputer hadir untuk mengantisipasi adanya kesenjangan teknologi komputer di kalangan masyarakat Indonesia pada umumnya dan masyarakat Jawa Tengah pada khususnya, dengan berbagai bidang usaha, meliputi:

- Hardware Trading
- Aftersales Service & Installation
- Education
- Publishing
- Data Processing
- Software Development
- Internet Services

Wahana Komputer berharap akan mampu memberikan secara menyeluruh untuk segala permasalahan yang berhubungan dengan teknologi komputer mulai dari aspek hardware, software, brainware, hingga integrasi sistemnya, sehingga pada akhirnya akan mampu berperan banyak dalam memperlancar bergulirnya era globalisasi dan informasi di Indonesia.

Didasari dengan misi dan visi yang jelas, Wahana Komputer berharap dapat membantu para peminat, pengguna, praktisi di bidang teknologi informasi untuk mendapatkan solusi dan informasi yang menyeluruh dan profesional, termasuk juga Anda para pembaca buku ini.

Walaupun dalam menyusun buku ini kami telah melakukan pengkajian dan penelitian yang mendalam serta berusaha untuk menyampaikan materi secara lengkap dan terstruktur, tetapi tentunya setiap karya tidaklah ada yang benar-benar sempurna sehingga mungkin buku ini kurang dapat memenuhi kebutuhan para pembaca, atau mungkin para pembaca masih mengalami kesulitan atau masalah setelah mempelajari buku ini.

Untuk itu, Wahana Komputer membuka lebar-lebar kesempatan bagi para pembaca pada khususnya dan masyarakat pengguna komputer pada umumnya untuk melakukan konsultasi mengenai berbagai kesulitan yang dihadapi khususnya mengenai apa yang telah dijelaskan di dalam buku ini melalui situs web kami atau via pos, fax, ataupun e-mail. Di samping itu para pembaca dapat memanfaatkan pula layanan pelatihan komputer kami melalui Lembaga Pendidikan Komputer Wahana untuk lebih mendalami materi-materi yang dipaparkan di dalam buku ini.

Wahana Komputer

Jl. MT Haryono 637 Semarang

Tlp. (024) 8314727, 8413238 fax. (024) 8413964

e-mail : info@wahanakom.com

web : www.wahanakom.com